

Endüstriyel Robot Kolları İçin Kolay Programlama Yöntemi Geliştirme

Sezgin Seçil

**YÜKSEK LİSANS TEZİ**

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Mayıs 2015

Development Of A Simple Programming Method For Industrial Robot Manipulators

Sezgin Seil

**MASTER OF SCIENCE THESIS**

Department of Electrical-Electronics Engineering

May 2015

# Endüstriyel Robot Kolları İçin Kolay Programlama Yöntemi Geliştirme

Sezgin Seçil

Eskişehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği Uyarınca  
Elektrik-Elektronik Mühendisliği Anabilim Dalı  
Kontrol ve Kumanda Sistemleri Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır

Danışman: Yrd. Doç. Dr. Metin ÖZKAN

Bu Tez ESOGÜ BAP tarafından “201315044(2013-77)” ve “201515A211(2015-731)”  
no’lu projeler çerçevesinde desteklenmiştir.

Mayıs 2015

## ONAY

Elektrik-Elektronik Mühendisliđi Anabilim Dalı Yüksek Lisans öğrencisi Sezgin Seçil'in YÜKSEK LİSANS tezi olarak hazırladığı "Endüstriyel Robot Kolları İçin Kolay Programlama Yöntemi Geliştirme" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

**Danışman** : Yrd. Doç. Dr. Metin ÖZKAN

**İkinci Danışman** : -

**Yüksek Lisans Tez Savunma Jürisi:**

**Üye** : Yrd. Doç. Dr. Metin ÖZKAN

**Üye** : Prof. Dr. Osman PARLAKTUNA

**Üye** : Prof. Dr. Abdurrahman KARAMANCIOĞLU

**Üye** : Yrd. Doç. Dr. Ahmet YAZICI

**Üye** : Yrd. Doç. Dr. Muammer AKÇAY

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve  
..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Hürriyet ERŞAHAN  
Enstitü Müdürü

## ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Yrd. Doç. Dr. Metin ÖZKAN danışmanlığında hazırlamış olduğum “Endüstriyel Robot Kolları İçin Kolay Programlama Yöntemi Geliştirme” başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 02/06/2015

Sezgin Seçil

İmza

## ÖZET

Endüstriyel robot kolları, üretim süreçlerinde yoğun olarak kullanılmaktadır. Bu robotlar üretim süreçlerinde kalite ve üretim hızının artmasını sağlarken, maliyetleri de önemli ölçüde düşürmektedir. Bundan dolayı büyük ölçekli seri üretim gerçekleştiren firmalarda robot kullanımını artmaktadır. Robot fiyatlarının azalması ve robotların ikinci el pazarının olması, küçük ölçekli firmaların da robot temin etmesini mümkün kılmaktadır. Ancak, temel sorun robotun temin edilmesinden çok, robotun programlanmasında ortaya çıkmaktadır. Robotun parçalar üzerinde işlem gerçekleştirebilmesi için operatör tarafından programlanması gerekmektedir. Robotun programlanması, her yeni parça için ve her yeni tanımlanan işlem için yapılması gerekmektedir. Bu durumda firmanın robotu temin ederken, programlamada uzman bir operatörü de istihdam etmesi gerekmektedir. Bu firmaya maliyetli gelmekte ya da operatör bulamamaktadır.

Bu tez çalışması kapsamında, robot kolunun programlamasını kolaylaştıracak ve böylece uzman operatör ihtiyacını giderecek yöntemlerin geliştirilmesi planlanmaktadır. Geliştirilen yöntemlerin kullanılmasıyla birlikte, robot tarafından bir parça üzerinde işlem gerçekleştirilebilmesi için gereken programlama birkaç adımda gerçekleştirilebilecektir. Önerilen sistemde öncelikle robot, 3-boyutlu görüntü oluşturabilen algılayıcıyı verilen parça etrafında dolaştırarak 3-boyutlu görselleştirmenin elde edilmesini sağlamak ve bu görseli operatöre sunmaktadır. Operatör, işlem gerçekleştirilecek noktaları görüntü üzerinde işaretlemektedir. Sistem tarafından otomatik olarak oluşturulan hareket planlamasıyla da robot işaretlenen noktalara hareket ederek işlem gerçekleştirmektedir. Aynı tipten başka parçalar üzerinde de oluşturulan hareket planıyla işlem gerçekleştirilebilmektedir. Böylece, robotun programlanması görüntü üzerinde yapılan basit işaretlemeyle sağlanabilmektedir.

Oluşturulan sistem, gerçek zamanlı bir uygulamayla test edilmiştir. Uygulama sonuçları, önerilen sistemin başarılı bir şekilde hedeflenen endüstriyel işlemi gerçekleştirebildiğini göstermektedir.

**Anahtar Kelimeler:** Endüstriyel robot kolu, robot programlama, 3-boyutlu modelleme, nokta bulutu, hareket planlama.

## SUMMARY

Industrial robot manipulators are used extensively in production processes. While improving the production speed and quality, these robots provides significantly reduction in costs. Therefore, the use of robots increases in the large-scale companies that perform mass production. Since the market prices of robots decreases and it is possible to buy used robots with a reasonable prices, a small-sized companies are able to provide a robot. However, the main problem is robot programming over obtaining a robot. The robot must be reprogrammed by an operator to perform operations on any parts. The robot programming should be performed for each new piece and each defined tasks. In this case, the company should employ a specialist operator once buy a robot. Employing a specialist operator is not cost-effective for the company or the company cannot find an operator.

In this thesis, it is intended to develop methods to facilitate the programming of the robot manipulator and thus to eliminate the need for specialist operators. With the use of methods developed, the robot may be automatically re-programmed in a several steps for performing operations on a part by the robot. Firstly, the robot navigate a range sensor around a given part; the resulting 3-D image is presented to the operator. The operator marks the points on the image where a process is performed by the robot. The robot motion planning process takes place automatically by the system such that the motion trajectory passes over the marked points. Then, the robot carries out the tasks by employing the motion plan. Another parts of the same type may be processed by the robot using the same motion plan. Thus, the robot programming task is performed by simple markings on the image.

Developed system is verified via a real-time application. The results of the application demonstrate that the proposed system fulfil the planned industrial process succesfully.

**Keywords:** Industrial Robot Manipulators, robot programming, 3-D modelling, point cloud, motion planning.

## TEŞEKKÜR

Bu tez çalışmamda; konunun belirlenmesinde, gerçekleştirilmesinde ve sonuçlandırılmasında her türlü bilimsel katkı ve manevi destek sağlayan, hiçbir konuda yardımlarını esirgemeyen, tecrübe ve görüşleriyle bana her zaman çözüm yolu gösteren danışman hocam Yrd. Doç. Dr. Metin ÖZKAN'a sonsuz teşekkürlerimi sunarım.

Çalışmalarım sırasında değerli görüş, katkı ve bilgilerini benden esirgemeyen hocam sayın Prof. Dr. Osman PARLAKTUNA'ya teşekkürlerimi sunarım.

Eğitim hayatım boyunca başarılarımı borçlu olduğum, maddi ve manevi desteklerini ve yardımlarını benden hiç esirgemeyen sevgili aileme minnet dolu sonsuz teşekkürlerimi sunarım.

Tezimin her aşamasında yanımda olan, dostluğu ve desteği için meslektaşım Arş. Gör. Gülin ELİBOL'a teşekkürlerimi sunarım.

ESOGÜ BAP tarafından 201315044(2013-77) ve 201515A211(2015-731) no'lu projeler kapsamında sağlanan maddi destek ve imkanlar için Eskişehir Osmangazi Üniversitesi kurumuna teşekkür ederim.



# İÇİNDEKİLER

## Sayfa

<b>ÖZET</b> .....	vi
<b>SUMMARY</b> .....	vii
<b>TEŞEKKÜR</b> .....	viii
<b>ŞEKİLLER DİZİNİ</b> .....	xi
<b>ÇİZELGELER DİZİNİ</b> .....	xiii
<b>SİMGELER VE KISALTMALAR DİZİNİ</b> .....	xiv
<b>1. GİRİŞ</b> .....	1
1.1. Amaç .....	2
1.2. Motivasyon.....	3
<b>2. LİTERATÜR ÖZETİ</b> .....	4
<b>3. SİSTEM DONANIMI</b> .....	9
3.1. Sistem Donanım Tasarımı .....	9
3.2. Endüstriyel Robot Kolu.....	15
3.3. Lazer Profil Algılayıcı.....	17
3.4. İş İstasyonu.....	20
3.5. Güç Kaynağı.....	21
3.6. Algılayıcı Montaj Aparatı .....	21
<b>4. SİSTEM YAZILIMI</b> .....	23
4.1. Robot Sunucu Modülü.....	23
4.1.1. Veri toplama amaçlı sunucu yazılımı .....	25
4.1.2. İşlem uygulama amaçlı sunucu yazılımı.....	25
4.2. Veri Toplama Modülü .....	27
4.2.1. Veri toplama işlemleri .....	27
4.2.2. Dönüşüm işlemleri.....	28
4.3. Görselleştirme ve İşlem Noktası Seçim Modülü.....	31
4.4. Robot İşlem Uygulama Modülü .....	32
4.4.1. Hareket planlama .....	35
4.4.2. ROS tabanlı hareket planlama .....	41

**İÇİNDEKİLER (devam)****Sayfa**

<b>5. UYGULAMALAR</b> .....	44
5.1. Yöntem Testi .....	44
5.1.1. Görselleştirme aşaması .....	45
5.1.2. İşlem noktası seçim aşaması .....	46
5.1.3. İşlem uygulama aşaması .....	49
5.2. Hareket Planlama Testi .....	55
5.2.1. Görselleştirme aşaması ve işlem noktası seçim aşaması .....	55
5.2.2. İşlem uygulama aşaması .....	57
<b>6. SONUÇLAR</b> .....	59
6.1. Sonuçlar .....	59
6.2. İleriye Yönelik Çalışmalar .....	60
<b>KAYNAKLAR DİZİNİ</b> .....	61
<b>EK AÇIKLAMALAR</b> .....	64
Ek Açıklamalar-A: Robot Urdf Model Dosyası .....	64

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
3.1. Sistem Donanım Tasarımı.....	10
3.2. Endüstriyel Robot Kolu Enerji ve Veri Bağlantıları.....	11
3.3. Robot Denetleyicisi Harici Çıkış Sinyal Kartı.....	12
3.4. Algılayıcı Besleme ve Tetikleme Devre Tasarımı.....	14
3.5. Kawasaki RS05L Robot Kolu.....	16
3.6. Kawasaki RS05L Çalışma Alanı.....	16
3.7. Kawasaki E70 Robot Denetleyicisi.....	17
3.8. Optik Üçgenleme Prensibi.....	18
3.9. Micro-Epsilon scanCONTROL 2900-100 Lazer Profil Algılayıcı.....	19
3.10. İş İstasyonu.....	20
3.11. Algılayıcı Montaj Aparatı (a) Görünüş (b) Teknik Çizim.....	22
4.1. Sistem Yazılım Tasarımı.....	24
4.2. Veri Toplama Amaçlı Sunucu Yazılımı Akış Diyagramı.....	26
4.3. İşlem Uygulama Amaçlı Sunucu Yazılımı Akış Diyagramı.....	26
4.4. Veri Toplama Modülünün Veri Toplama İşlemi Akış Diyagramı.....	28
4.5. Robot Denetleyicisinde Euler Açılı İfadeleri.....	29
4.6. Veri Toplama Modülünün Dönüşüm İşlemi Akış Diyagramı.....	30
4.7. Görselleştirme ve İşlem Noktası Seçim Modülü.....	31
4.8. Görselleştirme Üzerinde İşlem Noktası Seçim Örneği.....	32
4.9. MoveIt Yazılımının İçerdiği Ana Düğüm için Üst Seviye Sistem Mimarisi.....	34
4.10. Örnekleme Aşamasında Ağaç Üzerinde Örneklerin Oluşması.....	38
4.11. Örnekleme Aşamasında Hedef Durumun Ağaca Bağlanarak Aramanın Sonlandırılması Durumu.....	39
4.12. RRT-Connect Algoritması.....	40
4.13. Kawasaki RS05L İçin Oluşturulan Urdf Modeli.....	42
5.1. Uygulama için Kullanılan Endüstriyel Parça (a) Görünüş (b) Teknik Çizim.....	44
5.2. Taranan Parçaya Ait 3-Boyutlu Nokta Bulut Görselleştirmesi.....	45
5.3. Görsel Üzerinden Üçer Adet Nokta Seçilmesine İlişkin Görüntü.....	46

**ŞEKİLLER DİZİNİ (devam)**

<b><u>Sekil</u></b>	<b><u>Sayfa</u></b>
5.4. Büyük Çember Merkezlerini Saptamak İçin Kullanılan Algoritma .....	48
5.5. Simülasyon Ortamında Robot Kolu Modeli ve 3-B Nokta Bulut Verisi .....	49
5.6. Planlama Ortamına Yüklenen Nokta Bulut Verisi.....	50
5.7. Simülasyon Ortamında Endüstriyel İşlemin Gerçekleştirildiği Süreçte Sırayla Alınan Görüntüler.....	52
5.8. Endüstriyel İşlemin Gerçekleştirildiği Süreçte Sırayla Alınan Görüntüler.....	54
5.9. Hareket Planlama Testi İçin Ortaya Konulan Uygulama Ortamları (a) Engel İçermeyen (b) Engel İçeren .....	55
5.10. Hareket Planlama Testi İçin Elde Edilen Görselleştirme Sonuçları (a) Engel İçermeyen (b) Engel İçeren .....	56
5.11. Hareket Planlama Testi İçin Oluşturulan Hareket Planları (a) Engel İçermeyen (b) Engel İçeren.....	57
5.12. Hareket Planlama Testinin Engel İçeren Senaryosunun Gerçekleştirildiği Süreçte Sırayla Alınan Görüntüler .....	58

**ÇİZELGELER DİZİNİ****Cizelge****Sayfa**

3.1. Lazer Profil Algılayıcı Çok Fonksiyonlu Soketinde Yer Alan Pinler ve İşlevleri.....	13
3.2. RS05L Eklemlerinin Hareket Limitleri ve Maksimum Hızları.....	15
5.1. Uygulamada Seçilen Noktalar ile Hesaplanan Delik Merkezi ve Geçiş Noktası Koordinatları.....	47
5.2. Hareket Planlamanın Gerçekleştirildiği Nokta Dizisi.....	51
5.3. RRT-Connect Algoritmasının Planlama Parçalarında Üretilen Durum Sayısı ve Çözüme Ulaşma Süresi.....	53

## SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
$O$	İlk Rotasyonu İfade Eden Euler Açısı
$A$	İkinci Rotasyonu İfade Eden Euler Açısı
$T$	Üçüncü Rotasyonu İfade Eden Euler Açısı
$X$	Taban Koordinat Sisteminin Bir Ekseni
$Y$	Taban Koordinat Sisteminin Bir Ekseni
$Z$	Taban Koordinat Sisteminin Bir Ekseni
$x$	Hedef Koordinat Sisteminin Bir Ekseni
$y$	Hedef Koordinat Sisteminin Bir Ekseni
$z$	Hedef Koordinat Sisteminin Bir Ekseni
$X'$	İlk Rotasyon Sonucu Oluşan Koordinat Sisteminin Bir Ekseni
$Y'$	İlk Rotasyon Sonucu Oluşan Koordinat Sisteminin Bir Ekseni
$X''$	İkinci Rotasyon Sonucu Oluşan Koordinat Sisteminin Bir Ekseni
$X_{unp}$	X Ekseni İçin Robot Uç Noktası Pozisyonu
$Y_{unp}$	Y Ekseni İçin Robot Uç Noktası Pozisyonu
$Z_{unp}$	Z Ekseni İçin Robot Uç Noktası Pozisyonu

## 1. GİRİŞ

Endüstriyel alanda robot kollarının kullanımı, özellikle maliyetlerin düşürülmesi ve kalitenin artırılması amacıyla gün geçtikçe artmaktadır. Günümüz sanayisinde önemi artan endüstriyel robot kollarının kullanımı, genel olarak yoğun derecede hazırlık ve eğitim süreci geçirmesi gereken uzman operatör gerektirmektedir. Robot kolunun gerçekleştireceği işlemi denetleyen veya yöneten operatörün robot kolunun kullanımı üzerine alacağı eğitim yüksek maliyetlere sebep olabileceği gibi, bazen de yeni personel ihtiyacına dolayısıyla da ekstra maliyete sebep olmaktadır.

Lazer algılayıcı gibi çeşitli algılama sistemleri ile endüstriyel parçalara ait verilerin elde edilmesine yönelik çalışmalar da, endüstriyel alanda son dönemlerde önem kazanan çalışmalardandır. Algılayıcı sistemleri yardımıyla özellikle cismin 3-boyutlu modelinin elde edilmesine yönelik olan bu çalışmalar parça üzerinde istenen endüstriyel işlemin gerçekleştirilmesi sürecine yardımcı olmaktadır.

Bu tez çalışmasında, 6-eksenli endüstriyel robot kolu ve lazer profil algılayıcı kullanılarak endüstriyel robot kolları için otomatik programlama yöntemi geliştirilmektedir. Çalışmanın birinci bölümünde, bu tezde yapılan çalışmaya sebep olan motivasyon ve çalışmanın amacından bahsedilmektedir. İkinci bölümünde ise literatürde yer alan benzer türdeki çalışmalara değinilmektedir.

Üçüncü bölümde, geliştirilen sistemin içerdiği donanımlara ait bilgiler ile sistem donanım tasarımından bahsedilmektedir. Geliştirilen sistem, 6-eksenli endüstriyel robot kolu, lazer profil algılayıcı ve sistemin çalışmasına yönelik çeşitli elemanlar içermektedir. Bu bölümde öncelikle bu donanımlar arasında kurulan bağlantıların ve akışın yer aldığı donanım tasarımı anlatılmaktadır. Sahip olunan donanımlara ait özellikler ise bölümün devamında ele alınmaktadır.

Dördüncü bölümde, geliştirilen sistem için oluşturulan yazılım tasarımından, yazılımlardan ve bu yazılımların oluşturulmasında ele alınan işlemlerden bahsedilmektedir. İlk

olarak yazılımların birbirleriyle ilişkilerinin ve sistemdeki akışın anlatıldığı bu bölümde, daha sonra her bir yazılımdan ayrı ayrı bahsedilmektedir. Yazılımlara değinilen kısımda ilk olarak robotta yer alan sunucu yazılımını oluşturan veri toplama amaçlı sunucu yazılım ve işlem uyguluma amaçlı sunucu yazılım, ardından da veri toplama modülü ile bu modülün oluşturulmasında kullanılan Euler açılarına değinilmekte ve dönüşüm işlemlerinden bahsedilmektedir. Görselleştirme ve işlem noktası seçimini gerçekleştiren modül ve daha sonra robot işlem uygulama modülü ve bu modülün oluşumunda ele alınan hareket planlama algoritması ve ROS (Robot Operating System) üzerinde yapılan işlemler bu bölümün son kısmında ele alınmaktadır.

Beşinci ve altıncı bölümlerde sırasıyla geliştirilen sistem ile gerçekleştirilen örnek uygulamalar ve elde edilen sonuçlardan bahsedilmektedir. Ayrıca ileriye yönelik çalışmalara da yer verilmektedir.

## **1.1. Amaç**

Tez kapsamında, robotun programlamasını kolaylaştıracak ve böylece uzman operatör ihtiyacını giderecek veya azaltacak yöntemlerin ve beraberinde de sistemin geliştirilmesi amaçlanmaktadır. Geliştirilen yöntemlerin kullanılmasıyla birlikte, robot tarafından bir parça üzerinde işlem gerçekleştirilebilmesi için gereken programlama birkaç adımda gerçekleştirilebilmektedir.

Bu amaç için geliştirilen sistemde öncelikle, endüstriyel robot kolu aracılığıyla 3-boyutlu görüntü oluşturabilen algılayıcı endüstriyel parça etrafında dolaştırılarak, elde edilen 3-boyutlu görüntünün operatöre sunulması amaçlanmaktadır. Operatör işlem gerçekleştirilecek noktaları görüntü üzerinde işaretlemesiyle sistem tarafından otomatik olarak oluşturulacak hareket planı robotun işaretlenen noktalara hareket ederek işlemleri gerçekleştirmesi sağlanmaktadır. Böylece, robotun programlaması görüntü üzerinde yapılan basit işaretlemelerle gerçekleştirilebilmektedir.



## 1.2. Motivasyon

Endüstriyel robotlar, üretim süreçlerinde yoğun olarak kullanılmaktadır. Bu robotlar üretim süreçlerinde kalite ve üretim hızının artmasını sağlarken, maliyetleri de önemli ölçüde düşürmektedir. Bundan dolayı büyük ölçekli seri üretim gerçekleştiren firmalarda robot kullanımını artmaktadır. Robot fiyatlarının azalması ve robotların ikinci el pazarının olması, küçük ölçekli firmaların da robot temin etmesini mümkün kılmaktadır. Ancak, temel sorun robotun temin edilmesinden çok, robotun programlanmasında ortaya çıkmaktadır. Robotun parçalar üzerinde işlem gerçekleştirebilmesi için operatör tarafından programlanması gerekmektedir. Robotun programlanmasının, her yeni parça için ve her yeni tanımlanan işlem için yapılması gerekmektedir. Bu durumda firmanın robotu temin ederken, programlamada uzman bir operatörü de istihdam etmesi gerekmektedir. Bu firmaya maliyetli gelmekte ya da operatör bulamamaktadır. Endüstriyel alanda karşılaşılan bu problemler bu tezin motivasyonu olmaktadır.

## 2. LİTERATÜR ÖZETİ

Robot kol kullanılarak nesnelerin 3-boyutlu modelinin çıkarılması son zamanlarda popüler bir araştırma konusudur. Algılayıcı teknolojilerinin gelişmesi ile bu alanda önemli gelişmeler sağlanmıştır. Robot kolunun kullanımı ile elde edilen 3-boyutlu modeller çeşitli robotik uygulamalarında kullanılmaktadır. Robot tarafından çıkarılan 3-boyutlu model, engelden bağımsız hareket planlarının oluşturulması, parçaların teknik çizimlerinin elde edilmesi, parça üretim hatalarının saptanması gibi amaçlar için kullanılabilir.

Endüstriyel robot uygulamalarında, nesnelerin 3-boyutlu olarak algılanmasını kapsayan çeşitli çalışmalar bulunmaktadır. Larsson ve JAP (2006), nesnenin 3-boyutlu modelinin elde edilebilmesi için dört adımda gerçekleştirilen bir teknik geliştirmiştir. Bu teknik, şu adımları içermektedir: i) Nesneye bakış açılarını planlamak, ii) fiziksel olarak algılayıcının, nesneye göre pozisyon ve oryantasyonunu belirlemek, iii) taramaları gerçekleştirmek, iv) elde edilen geometrik verileri ortak bir referans koordinat sistemi için bir araya getirip veri tekrarlamasız (non-redundant) bir model oluşturmak. Ortaya koydukları bu yöntemin sonuçlarının endüstriyel robotların hareket planlamasında kullanılabileceği belirtilmiştir.

Bir başka çalışmada Nayak vd. (2003), bir kaynak robotu için kaynak işlemindeki vakit kaybını aza indireyecek bir yöntem ortaya koymuşlardır. Ortaya konulan yöntemle göre kaynak yapılacak yolun önceden programlanarak belirlenmesiyle ortaya çıkan zaman kaybının ortadan kalkacağı belirtilmiştir. Bu bağlamda kaynak işlemini gerçekleştirecek olan uç, gerçek zamanlı ve 3-boyutlu olarak toplanan ve işlenen kaynak yolu bilgisine göre hareket edecektir. Önerilen yöntem için yapılan testler sonucunda kaynak uygulamalarında gelişmeler gözlemlenmiştir.

Bruccoleri vd. (2007), bir kaynak robotu için çevrimdışı programlama yaklaşımı geliştirmişlerdir. Geliştirdikleri yöntemle kullanıcının robotun çalışma alanındaki elemanları ayarlamasını ve gerçekleştirilecek hareketleri grafik tabanlı olarak belirlemesini mümkün kılmayı amaçlamışlardır. Kullanıcı tarafından yapılan belirlemeler sonrasında ise robot için

gerekli hareket ve çalışma sıraları tanımlanmaktadır. Yapılan testler sonucu ilgili metot ile yüksek çeşitlilikteki parçaların yüksek derecede esneklikle işlenebildiğini göstermişlerdir.

Son vd. (2002), ortaya koydukları çalışmada serbest forma sahip bir yüzeyin tersine mühendisliği (reverse engineering) için otomatik bir tarama sistemi geliştirmişlerdir. Geliştirdikleri sistem, en uygun bir tarama planı üretmektedir. Bu plan, sistemde yer alan ekipmanların çeşitli parametrelerini de göz önünde bulundurarak gerekli tarama sayısını, tarama sırasında bakış açılarını ve tarama sırasında izlenecek güzergâhları içermektedir.

Luyi (2011), üç boyutlu olarak algılanacak modelin oryantasyonunun belirlenmesi üzerine dört adımlık bir metot geliştirmiştir. Bu adımlar şu şekildedir: i) Standard parça üç boyutlu tarama sistemi ile taranır ve standart parçanın taranmış modeli elde edilir, ii) üzerinde çalışılmakta olan parça üç boyutlu tarama sistemi ile taranır ve üzerinde çalışılmakta olan parçanın taranmış modeli elde edilir, iii) elde edilmiş iki model Özyineli En Yakın Nokta algoritması kullanılarak yüzeyleri çakışacak şekilde yerleştirilmek üzere kaydedilir, iv) robot kontrol programı elde edilenlere göre güncellenir. Bu yöntem, çevrimiçi kontrol ve kalite ölçme süreci için robotik tutkallama sisteminde kullanılmıştır.

Robot koluyla gerçekleştirilen profil taramaları ve endüstriyel uygulamalarda hareket yörüngelerinin ortaya konulması verimlilik, hareket kabiliyeti ve engellerden sakınma gibi konuları içerdiğinden bu yönleriyle birçok çalışmaya konu olmaktadır. Larsson ve Kjellander (2007), yaptıkları çalışmada lazer profil tarayıcısı ve robot kolu kullanarak var olan birçok yöntemden farklı bir yol izlemişlerdir. Geliştirdikleri sistem, var olan yöntemlerin aksine birçok açıdan hatta kıvrımlı yollarda dahi taramayı mümkün kılmaktadır. Sistem, ölçme sisteminin bilgisayar destekli tasarım yazılımı tarafından kontrol edilmesiyle gerçekleştirilmektedir. Bu sayede kıvrımlı hareket yörüngelerinde dahi ölçme kapasitesine ulaşılmaktadır.

Son vd. (2003), çalışmalarında çoklu parçalara sahip karmaşık yüzeyleri taramak için hareket yörüngeleri hakkında yeni bir metot önermişlerdir. Metot, sezgilerle üretilmesi zor olan ölçme planını ortaya koymaya yönelik içeriğe sahiptir. Buna göre tarama sayısı, tarama yönü ve tarama hareket yörüngesi metot tarafından hesaplanmaktadır. Tarama planının üretilmesi için gerekli olan işlem bilgisayar destekli tasarım modelinin otomatik olarak elde

edilmesiyle başlamaktadır. Tasarım bilgisi, işlevsellik ve ilgili parametreler etkileşimli bir biçimde sağlanmaktadır. Elde edilen bilgiyle orijinal yüzey modeli parametre kümesinde örneklenir, ayrıca örneklenmiş noktalar tarayıcının açısına göre gruplanır. Tarama yönü, bir gruba ait olan örneklenmiş noktaların global ortalama vektörünün hesaplanmasıyla belirlenir. Her bir tarama yönü için optik tarama hareket yörüngeleri ise örneklenmiş noktaların iki boyutlu düzleme yansıtılması ile hesaplanır. Optik kısıtlar ise üretilmiş hareket yörüngesini garanti etmek için doğrulanır.

Jeon vd. (1993), hareket yörüngelerinin planlanması için verimli ve hareket kabiliyeti yüksek bir teknik önermişlerdir. Önerilen teknik iki adım içermektedir ve bu adımlar şu şekildedir: i) İstenen mesafe, maksimum hız ve maksimum ivme ortaya konarak trapezoid hız profili üretilir, ii) daha düz bir trapezoid hız profili, daha önce üretilen hız profilinin kullanımıyla her örnekleme anında kendisine karşılık gelen oranla çarpılarak elde edilir. Bu tekniği kullanarak hızlanma karakteristiğinin yavaşlama karakteristiğinden farklı olduğu düz bir hız profili elde etmişlerdir. Böylelikle hareket süresini kısaltırken, titreşimi de azalttıklarını ortaya koymuşlardır.

Wurll ve Henrich (2001), çalışmalarıyla endüstriyel robotlar için çarpışma önlenmesini de içine alacak şekilde yol planlamasına katkıda bulunmuşlardır. Yaklaşımları dahili ve ayrıklaştırılmış konfigürasyon uzaylarında çalışmakta ve çarpışmalar ise Kartezyen çalışma alanında hiyerarşik uzaklık hesaplamasıyla ortaya çıkarılmaktadır. Yöntemleri A\* arama algoritmasını temel almakta ve önemli bir çevrimdışı hesaplamaya gerek duymamaktadır. Çalışmalarıyla kurulum maliyetlerini düşürebilecek bir adım atmışlar ve noktadan noktaya ile çoklu hedef yol planlaması için toplam sıra yol uzunluğunun azaldığı sonuçlar elde etmişlerdir.

Literatürde nesnelerin görselleştirilmesine yönelik de çalışma birçok çalışma bulunmaktadır. Bu çalışmalar veri işlemeye yönelik metodlar ve veri toplama sistem tasarımları olmak üzere iki şekilde sınıflandırılabilir. Lazer algılayıcı ve koordinat ölçüm cihazları (CMM) 3-boyutlu yüzey veri toplama sistemleri için karşılaşılan tasarımlarda kullanımına en çok başvuru alan algılayıcılarıdır.

Chan vd. (2000), yaptıkları çalışmada CCD kamera, 3-boyutlu lazer sayısallaştırıcı ve CMM kullanmışlardır. Çalışmalarında yüzey tarama ve tersine mühendislik için katı model oluşturmaya yönelik bir çalışma ortaya koymuşlardır. Lazer tarayıcının takip edeceği yolu izlemesi için çiftli görüntülere bir sinir ağı tabanlı bölütleme algoritması uygulamışlardır. Sonuç olarak otomatik bir şekilde CAD/CAM paketleri için de kullanılabilir nesne yüzey bilgisini üreten süreci ortaya koymuşlardır. Xie vd. (2005), karmaşık yapıdaki nesnelerin taranması için 5-eksenli lazer tarama sistemi geliştirmişlerdir. Geliştirdikleri sistem, lazer ışın algılayıcısı, döner başlık ve CMM içermektedir. Elde ettikleri lazer algılayıcı verisi tek boyutlu olduğundan, genel koordinat sistemine göre de üç boyutlu veri ifade edebilmek için koordinat dönüşümü için bir sistem modeli ortaya koymuşlardır. Çalışmalarının sonucu olarak sistemlerinin bir parçayı herhangi bir yönden tarayarak düşük hassasiyetli hesaplamalar yapabildiğini göstermişlerdir.

Bunun yanısıra algılayıcıların hem obje etrafına statik olarak yerleştirildiği, hem de manuel olarak obje üzerinde dolaştırıldığı veri toplama yapıları da literatürde yer almaktadır. Tognola vd. (2003), çalışmalarında biyomedikal uygulamalar için anatomik kısımların 3-boyutlu rökonstrüksiyonunu gerçekleştirmek adına bir teknik ortaya koymuşlardır. Önerilen tarama sistemi He-Ne lazer, iki CCD kamera ve gerçek zamanlı video işlemcisi içermektedir ve obje yüzeyinin lazer ışınıyla manuel olarak taranmasını içermektedir. Benzer olarak bazı veri toplama yapıları bir kamera ve projektör içermektedir. Lanman vd. (2009), bir objenin 3-boyutlu şeklini elde etmek için ortografik projektör ve bir çift düzlemsel ayna kullandıkları bir çalışma ortaya koymuşlardır. Bu çalışma, tek yapılı ışık dizisi kullanılarak nesnelerin 360° rökonstrüksiyonunu elde etmeye yöneliktir. Ortaya koydukları sonuca göre geliştirdikleri sistem, birden fazla taramayı birleştirme veya tarayıcıların sayıca çoğaltılması ihtiyacını ortadan kaldırmaktadır. Zexiao vd. (2007), yaptıkları çalışmada büyük parçaların hassas ve tersine mühendislik için uygun olacak bir şekilde 3-boyutlu ölçümü için bir sistem ortaya koymuşlardır. Önerdikleri sistem CCD kamera, galvanometre ve bir lazer projektör içermektedir.

Yapıların görselleştirilmesi, sualtı araştırma çalışmaları gibi alanlar için de cazip bir araştırma konusu olmaktadır. Papadopoulou vd. (2011), güçlü bir lazer tarayıcı ve sonar algılayıcı kullanımıyla geliştirdikleri robotik ve yazılım sistemi sayesinde denizcilikle ilgili

yapıların su üstünde ve altında kalan kısımlarını taramaya yönelik bir çalışma sunmuşlardır. Sunulan çalışma, deniz ortamında 3-B yüzey rökonstrüksiyonuyla ilgili problemlerin çözümüne odaklanmıştır. Önerdikleri sistem, LIDAR ve bir deniz botu altına monte edilmiş sonar algılayıcı içermektedir. Çalışmalarının sonucu olarak nesnenin düşük kalitede haritasını ortaya çıkarmışlardır.

Bazı çalışmalar ise sadece veri işleme metodları üzerine odaklanmaktadır. Karbacher vd. (2001), eğimli veya düz üçgenler ağına dayanan ve yüzey tanımlaması üretecek neredeyse otomatik bir metot sunmuşlardır. Çalışmaları, optik bir 3-boyutlu algılayıcı kullanımıyla veri toplama işlemini içermekte olup örgü azaltılması, düzleştirme ve kayıp verinin yeniden yapılandırılması işlemlerine değinmektedir. Önerdikleri metodun bir parçası olarak üretilen örgülerin görselleştirme ve prototipleştirme için uygun olduğunu ortaya koymuşlardır.

Önerdiğimiz sisteme benzer olarak ise veri toplama yapıları için robot kolu içeren 3-boyutlu görselleştirme sistemlerine de literatürde rastlanmaktadır. Vincenzo vd. (2010), gerçek zamanlı şekil elde edimi için lazer ışını ve web kamerası kullanımına dayanan bir lazer tarayıcı sistemi ortaya koymuşlardır. Çalışmaları, özel tasarıma sahip 3-eksenli bir robot kolunun hedef obje üzerinde dolaşarak cismin taranmasını içermektedir. Hem düzlemsel, hem de düzlemsel olmayan yüzeylerde yaptıkları taramalarda önerdikleri metodun robotik uygulamalar için kullanılabileceğini göstermişlerdir.

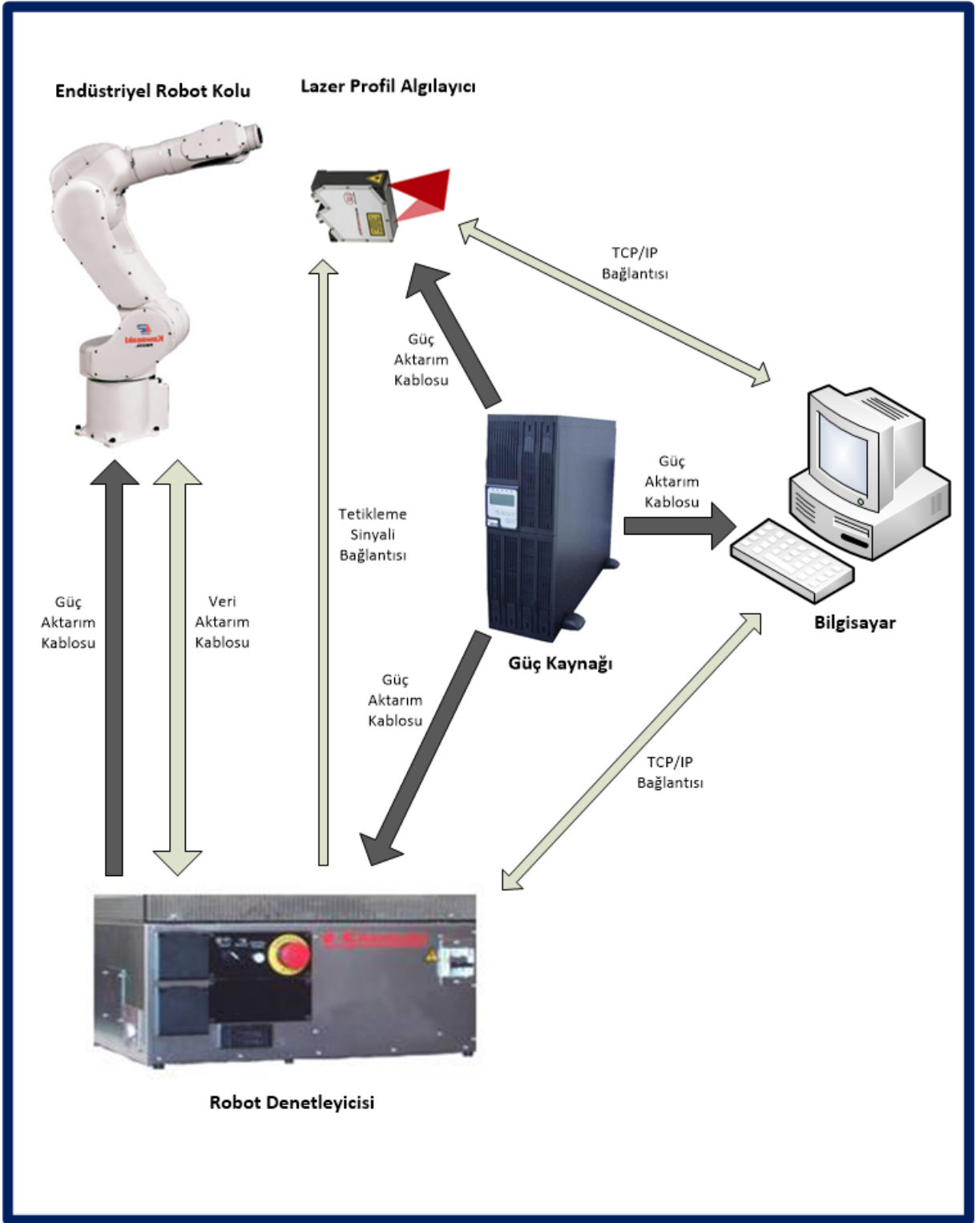
### 3. SİSTEM DONANIMI

Geliştirilen sistem, hedeflenen amaca ulaşabilmek için donanımsal olarak 6-eksenli endüstriyel robot kolu ve lazer profil algılayıcı gibi iki temel bileşen içermektedir. Bunun yanında, çalışma ortamının ve gerçekleştirilecek sürecin işlevselliği için iş istasyonu, güç kaynağı ve algılayıcı montaj aparatı gibi yardımcı donanımsal bileşenler de geliştirilen sistemde yer almaktadır. Ortaya konulan sistem donanım tasarımı, tüm donanım bileşenlerinin sisteme uygun bir şekilde entegre olmuş halini yansıtmaktadır.

#### 3.1. Sistem Donanım Tasarımı

Geliştirilen sistem için oluşturulan sistem donanım tasarımı, donanımlar arasındaki bağlantıları ve sistemin donanımsal işleyişini gerçekleştirmektedir. Geliştirilen sistem, endüstriyel robot kolu, lazer profil algılayıcı ve işleyişe yardımcı donanımsal elemanlar içerdiğinden bu tasarımla bir düzen ortaya konmaktadır. Mevcut elemanlar arasında sinyal gönderimi, enerji aktarımı ve veri paylaşımı gibi işlemler gerçekleştiğinden böyle bir tasarıma gerek duyulmaktadır. Oluşturulan tasarıma ait genel akış Şekil 3.1'de gösterilmektedir.

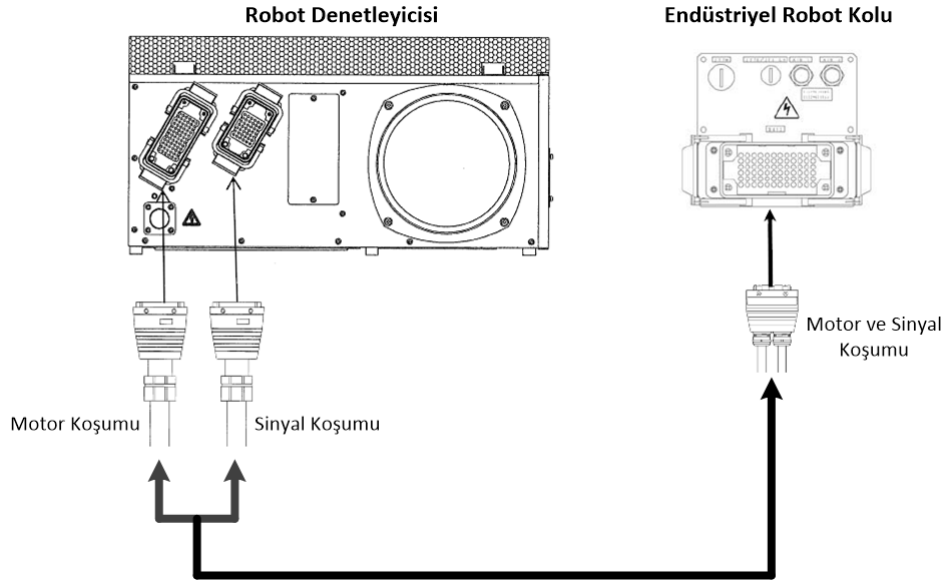
Sistemin sahip olduğu donanımlara enerji, kesintisiz güç kaynağı üzerinden dağıtılmaktadır. Endüstriyel robot kolu gereken enerjiyi, kesintisiz güç kaynağıyla beslenen robot denetleyicisi üzerinden sağlamaktadır. Sistemde yer alan bilgisayar ise enerjiyi direkt olarak güç kaynağından almaktadır. Lazer profil algılayıcının çalışma gerilimi, kesintisiz güç kaynağının çıkış gerilimine uygun olmadığı için özel bir devre tasarımına ihtiyaç duysa da gerektirdiği enerji yine kesintisiz güç kaynağı tarafından sağlanmaktadır.



Şekil 3.1: Sistem Donanım Tasarımı

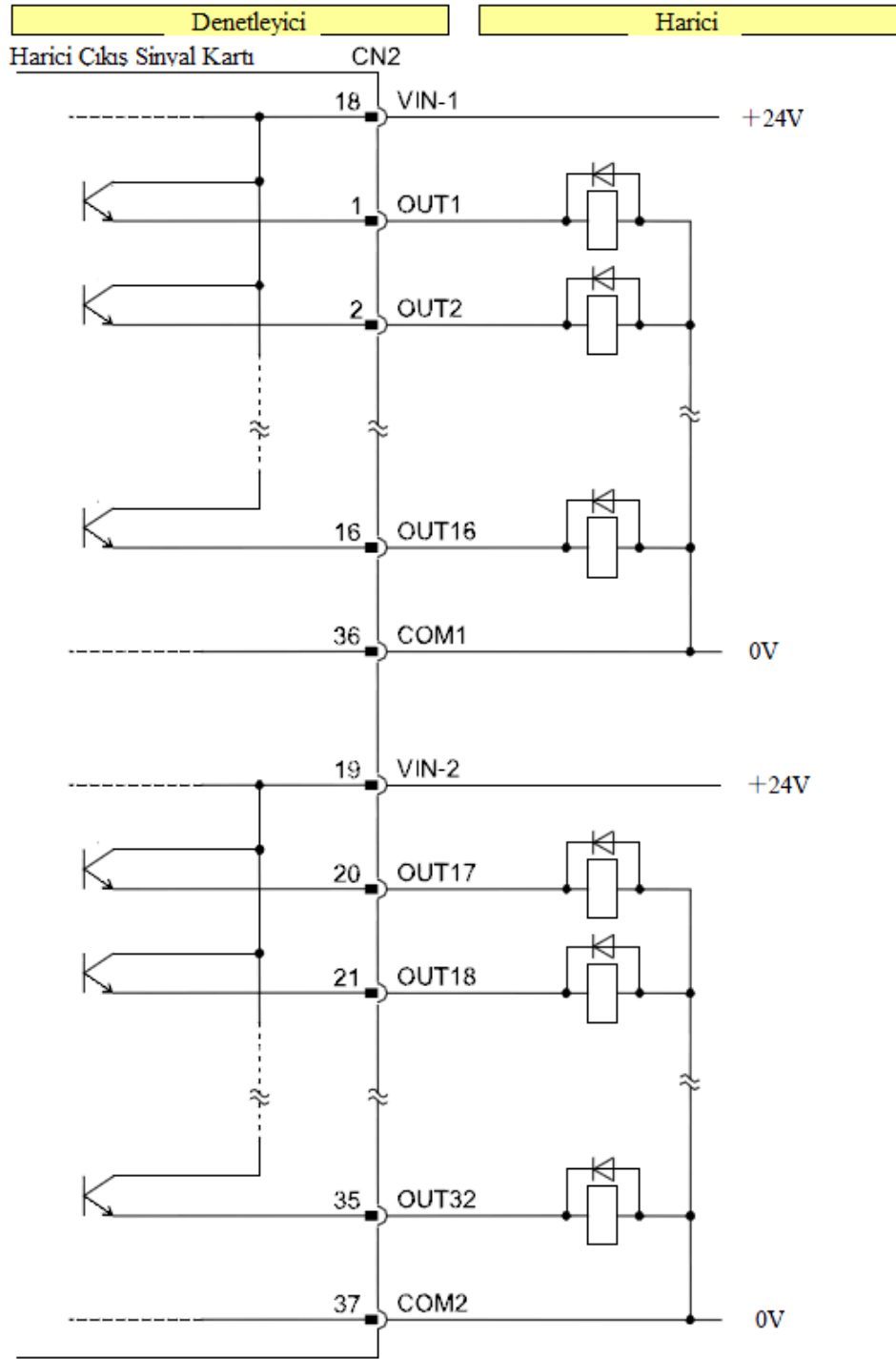


Endüstriyel robot kolu, enerji ihtiyacını robot denetleyicisi üzerinden karşılamaktadır. Aynı zamanda veri alışverişi için de denetleyici ile arasında bağlantı kurulmaktadır. Bu iki bağlantı, üretici firmanın tarafından üretilen ve motor ile sinyal koşumları adı verilen kablolarla sağlanmaktadır. Bu bağlantılar Şekil 3.2’de gösterilmektedir.



Şekil 3.2: Endüstriyel Robot Kolu Enerji ve Veri Bağlantıları (Kawasaki Heavy Industries, 2015)

Lazer profil algılayıcının, ortaya konulan sistem donanım tasarımına göre kesintisiz güç kaynağı tarafından enerjisi sağlanırken, robot denetleyicisi tarafından tetikleme sinyali bağlantısı aracılığıyla tetiklenmesi gerekmektedir. Lazer profil algılayıcının çalışma geriliminin ve kesintisiz güç kaynağının çıkış geriliminin farklılık göstermesinin yanında robot denetleyicisinin sahip olduğu harici çıkış sinyali kartının çıkışları, lazer profil algılayıcının çok-fonksiyonlu soketinin girişleri için uygun olmadığından ayrı bir devre tasarımına gerek duyulmaktadır. Öyle ki, robot denetleyicisinin sahip olduğu harici çıkış sinyali kartının çıkışları Şekil 3.3’de gösterildiği gibi PNP tipi transistör çıkışıdır.



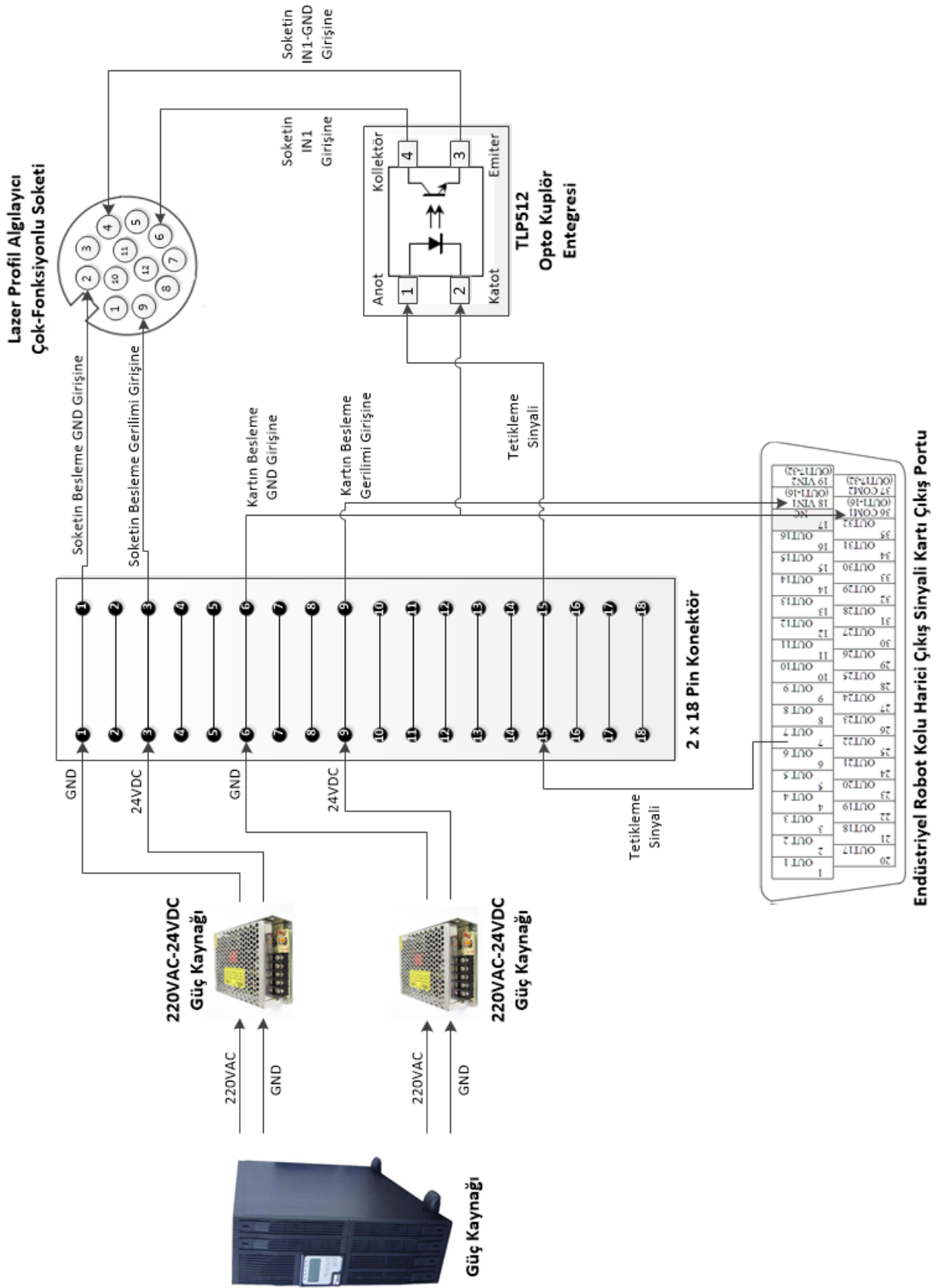
Şekil 3.3: Robot Denetleyicisi Harici Çıkış Sinyal Kartı (Kawasaki Heavy Industries, 2015)

Lazer profil algılayıcı, tetikleme sinyalini çok-fonksiyonlu soketi aracılığıyla almaktadır ve bu soketin girişlerine PLC, NPN tipi transistör veya kontak (switch) türünden girdi

sağlayan kaynaklar bağlanması gerektiğinden yeni bir devre tasarımı ortaya koyulmaktadır. Bu amaç için oluşturulan devre tasarımı, opto-kuplör devresi içermektedir ve dolayısıyla algılayıcının çok-fonksiyonlu soketine kontak türünde girdi sağlanmaktadır. Oluşturulan devre tasarımının içerdiği iki adet 220VAC-24VDC türünden güç kaynağı ise hem algılayıcının gerektirdiği enerjiyi çalışma gerilimine uyarak sağlamakta, hem de robot denetleyicisinin harici çıkış sinyali kartı için gerekli harici besleme gerilimini sağlamaktadır. Oluşturulan devre tasarımına ait şematik Şekil 3.4’de gösterilmektedir. Lazer profil algılayıcının sahip olduğu çok fonksiyonlu sokette yer alan pinler ve işlevleri ise Çizelge 3.1’de gösterilmektedir.

Çizelge 3.1: Lazer Profil Algılayıcı Çok Fonksiyonlu Soketinde Yer Alan Pinler ve İşlevleri

Pin	İşlev
1.	- Lazer açma/kapama
2.	GND (24V)
3.	+ Lazer açma/kapama
4.	GND - IN1
5.	Giriş - IN2
6.	Giriş - IN1
7.	GND - IN3
8.	GND - IN2
9.	Besleme (24V)
10.	Giriş - IN3
11.	/RS422
12.	RS422



Şekil 3.4: Algılayıcı Besleme ve Tetikleme Devre Tasarımı

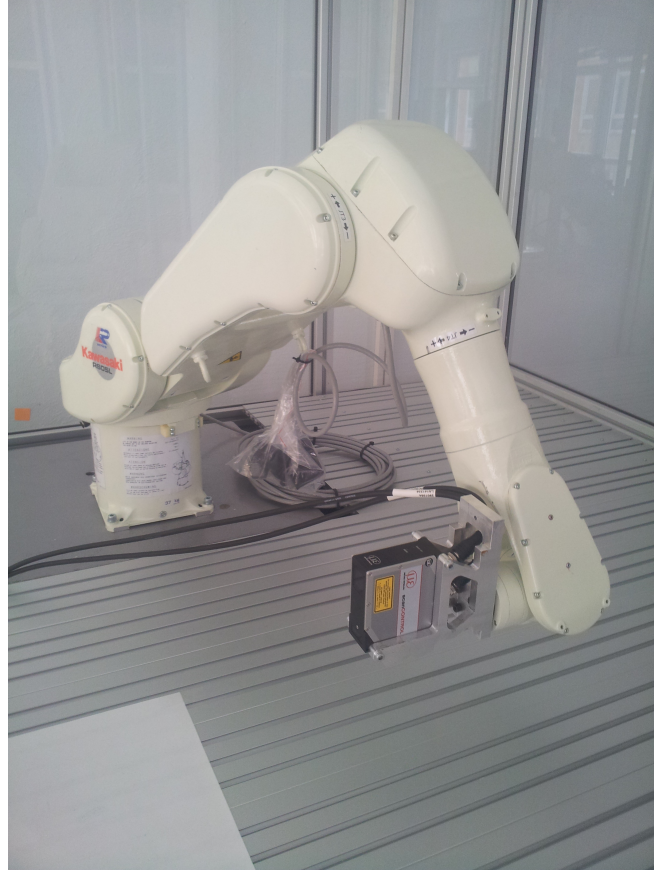
### 3.2. Endüstriyel Robot Kolu

Endüstriyel robot kollarının endüstride kullanımına kaynak, boyama, parçaların montajı gibi üretim safhalarında sıkça başvurulmakta ve bu robotlar kullanımlarıyla üretime kalite ile maliyet bakımından olumlu katkıda bulunmaktadır. Ortaya koyulan sistemde de üzerinde çalışılan robot kolu, bu tarz üretim safhalarında görev alabilecek kapasitede 6-eksenli endüstriyel bir robot koludur.

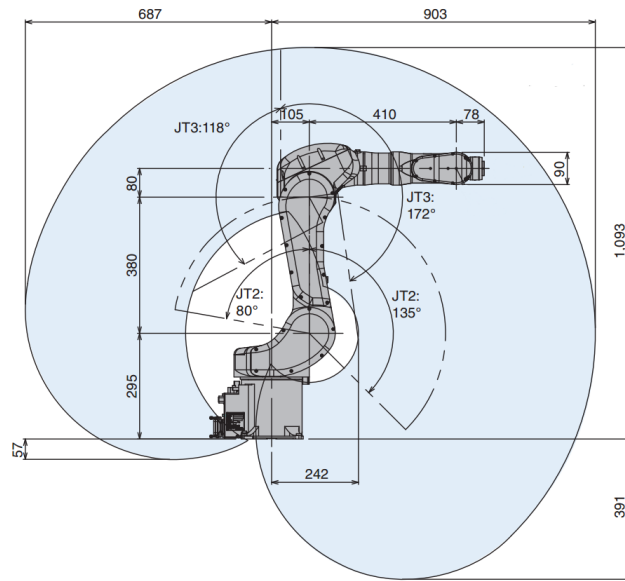
Bu tez çalışmasında kullanılan 6-eksenli endüstriyel robot kolu, Kawasaki firması tarafından üretilen RS05L model endüstriyel robot koludur. RS05L, Kawasaki firmasının ürettiği R serisi robot kollarından biri olup, küçük ve orta yükte endüstriyel işlemleri gerçekleştirebilecek kapasitededir. 6 eklemlili bir endüstriyel robot kolu olan RS05L'in, nominal yük taşıma kapasitesi 5 kg, tekrarlanabilirliği 0.03m, maksimum hızı ise 9300 mm/s'dir. Montaj, dağıtım, denetleme, makine destek, malzeme taşıma, kaynak yapma gibi uygulamaları olan RS05L'in zemin, tavan ve duvar gibi bölümlere kurulması mümkündür. Bu tez çalışmasında RS05L, zemin üzerine monte edilmiş şekilde bulunmaktadır. Oluşturulan sistemde zemine monte edilmiş şekilde hazır bulunan Kawasaki RS05L'e ait bir görüntü Şekil 3.5'de gösterilmektedir. RS05L robot kolunun çalışma alanında yatayda erişimi 903 mm, dikeyde erişimi 1484 mm'dir. RS05L'nin çalışma alanı Şekil 3.6'de gösterilmektedir. Her bir eklemin hareket limitleri ve bu eklemler için maksimum hızlar üretici firma tarafından yazılımsal olarak belirlenmiştir ancak bu limitler yazılımsal olarak, belirlenen aralığın dışına çıkmamak kaydıyla, değiştirilebilmektedir. Eklemlerin hareket limitleri ve maksimum hızları Çizelge 3.2'de gösterilmektedir.

Çizelge 3.2: RS05L Eklemlerinin Hareket Limitleri ve Maksimum Hızları

Eklemler	Hareket Aralığı	Maksimum Hız
1.	$\pm 180^\circ$	300 °/s
2.	$+135^\circ \sim -80^\circ$	300 °/s
3.	$+118^\circ \sim -172^\circ$	300 °/s
4.	$\pm 360^\circ$	460 °/s
5.	$\pm 145^\circ$	460 °/s
6.	$\pm 360^\circ$	740 °/s



Şekil 3.5: Kawasaki RS05L Robot Kolu



Şekil 3.6: Kawasaki RS05L Çalışma Alanı (Kawasaki Heavy Industries, 2015)

RS05L robot kolu, Linux tabanlı işletim sistemi kullanan Kawasaki E70 adlı robot denetleyicisi ile kontrol edilmektedir. Kawasaki E70 robot denetleyicisi Şekil 3.7'da gösterilmektedir. Sistemde RS05L'yi elle kontrol etmek veya öğretme yoluyla kontrol etmek için ayrıca robot denetleyicisine bağlanan kumanda paneli de bulunmaktadır. Daha detaylı ve derinlemesine kontrol için ise robot hareketlerinin programlanması adına üretici firma tarafından özel olarak geliştirilen ve "AS" adı verilen programlama dili kullanılmaktadır. E70 robot denetleyicisi ile veri alışverişi RS232 ve Ethernet iletişim protokolleriyle sağlanabilmektedir. Aynı zamanda robot denetleyicisi genel amaçlı harici sinyaller için 1TW adı verilen bir kart üzerinde giriş ve çıkış portları bulundurmaktadır. Bu tez çalışmasında robot denetleyicisi ile bilgisayar arasındaki iletişim için Ethernet iletişim protokolüne başvurulmaktadır. Bunun yanında robot denetleyicisinin sahip olduğu 1TW kartı üzerindeki çıkış portu algılayıcıya tetikleme sinyali gönderilmesi amacıyla kullanılmaktadır.

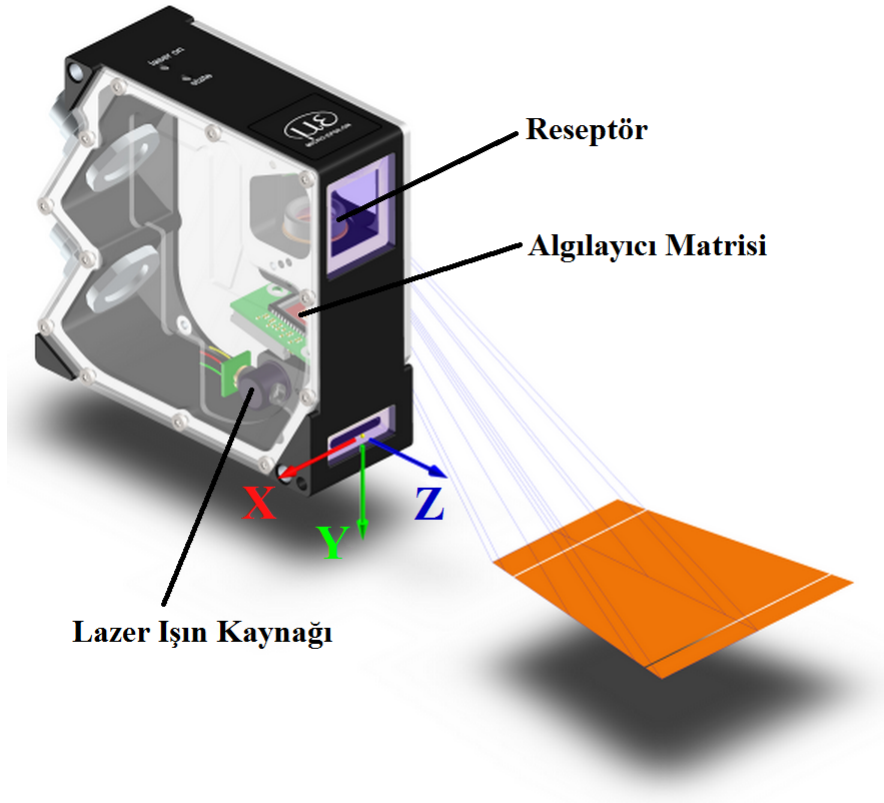


Şekil 3.7: Kawasaki E70 Robot Denetleyicisi

### 3.3. Lazer Profil Algılayıcı

Lazer profil algılayıcılar, görülebilir lazer ışınının hedef yüzey üzerinde çizgi oluşturacak şekilde yüzeye düşürülmesiyle ölçüm yapan algılayıcılardır. Lineer optik bir sistem tarafından yüzeye düşürülen ışığın yansıması yüksek kalitede optik sistemin içerdiği

alıcı sensör matrisi tarafından 2-boyutlu olarak hesaplanmaktadır. Lazer profil algılayıcıların gerçekleştirdiği bu ölçüm optik üçgenleme prensibine dayanmaktadır. Bu tür algılayıcılarda uzaklık bilgisinin yanında (z-ekseni), lazer çizgisi üzerindeki her bir noktanın esas konumu (x-ekseni) da hesaplanmaktadır. Lazer profil algılayıcının, optik üçgenleme prensibine dayanan ölçümü Şekil 3.8’de gösterilmektedir. Optik üçgenleme prensibine göre kaynaktan çıkan lazer ışını, lineer optik bir sistem sayesinde hedef yüzeye çizgi halinde yansıtılır, yüzeydeki lazer ışın çizgisinden yansıyan ışık ise reseptör tarafından algılayıcı matrisi üzerine düşürülür ve 2-B olarak ölçüm yapılır.



Şekil 3.8: Optik Üçgenleme Prensibi (Micro-Epsilon, 2015)

Lazer profil algılayıcılar için genel itibariyle ölçüm aralığı, doğrusalılık, çözünürlük ve ölçüm oranı gibi dört temel özellikten bahsedilebilmektedir. Bu özelliklerden ölçüm ara-



lığı, hedef yüzey üzerinde güvenilir ölçüm yapabilmek için algılayıcı uç noktasının bulunması gereken mesafe aralığıdır. Ölçüm aralığı başlangıç noktası ve bitiş noktası olmak üzere iki değerle ifade edilebilmektedir. Doğrusallık ise ölçüm aralığı üzerindeki en uygun düz doğrudan en yüksek sapma olarak tanımlanır. Çözünürlük, sensörün ölçüm aralığındaki değişimde saptayabileceği en küçük artış iken ölçüm oranı, algılayıcıdan verilerin elde edilme hızını anlatmaktadır. Bu tez çalışmasında kullanılan lazer profil algılayıcı, Micro-Epsilon firması tarafından üretilen scanCONTROL 2900-100 model lazer profil algılayıcıdır. ScanCONTROL 2900-100, Micro-Epsilon firmasının ürettiği scanCONTROL 2900 ailesi algılayıcılarından biri olup, yüksek çözünürlük ve yüksek profil frekansı gerektiren endüstriyel işlemleri gerçekleştirebilecek kapasitededir. Profil frekansı 4000Hz'e kadar çıkabilen scanCONTROL 2900-100'ün ölçüm oranı 2560000 nokta/sn, z ekseni ölçüm aralığı 265 mm'ye kadar, x ekseni ölçüm aralığı 143,5 mm'ye kadar, z ekseni referans çözünürlüğü  $2 \mu\text{m}$ , x ekseni çözünürlüğü 1280 noktaya kadar ve doğrusallığı  $\% \pm 0.16$  kadardır. Micro-Epsilon scanCONTROL 2900-100 lazer profil algılayıcı Şekil 3.9'de gösterilmektedir.

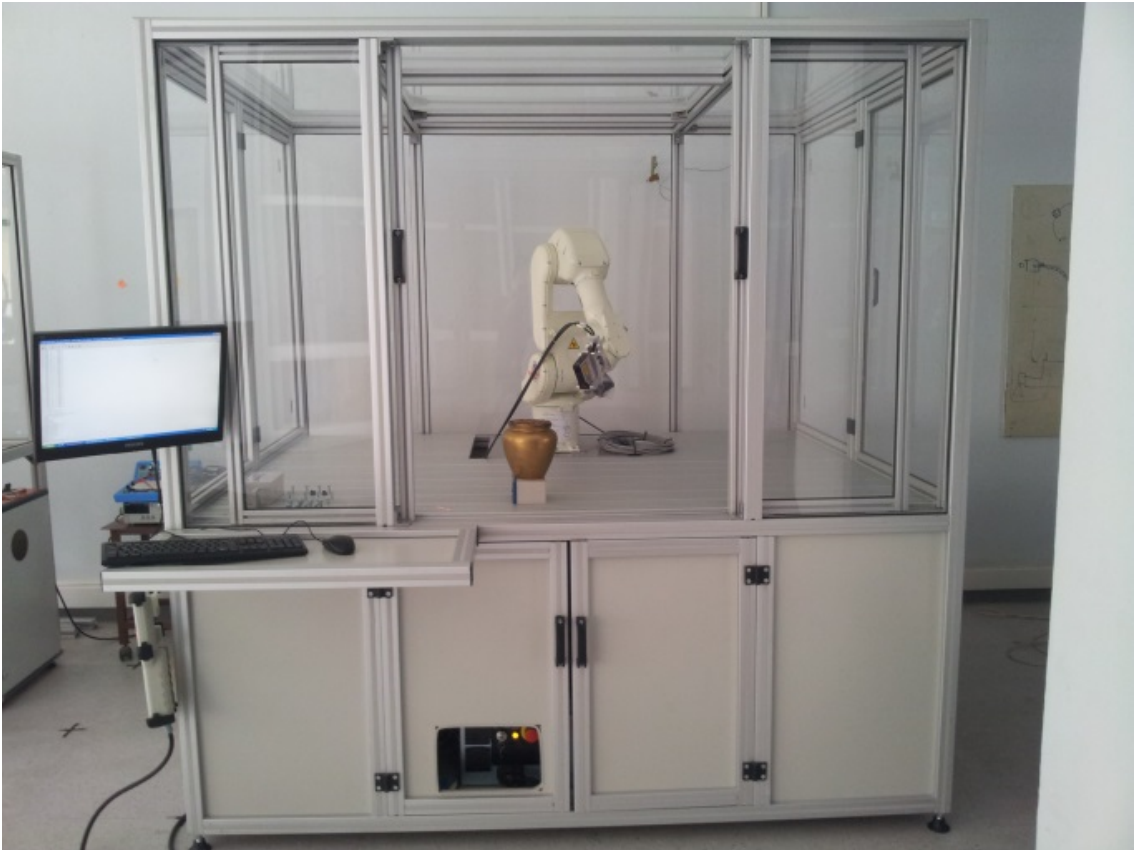


Şekil 3.9: Micro-Epsilon scanCONTROL 2900-100 Lazer Profil Algılayıcı (Micro-Epsilon, 2015)

Lazer profil algılayıcı ile veri alışverişi RS422 ve Ethernet iletişim protokolleriyle sağlanabilmektedir. Aynı zamanda lazer profil algılayıcı enerji ihtiyacını ethernet üzerinden sağlamaktadır.

### 3.4. İş İstasyonu

Oluşturulan sistem, sistem elemanlarının güvenli ve dış etkenlerden korunacak şekilde çalışabilmesi ile üzerinde çalışılacak parçaların sabitlenmesi amaçlarının yanında robot kolunun yerleştirildiği tezgahı içeren ve sistemde yer alan bilgisayar gibi elemanları bünyesinde barındıran bir iş istasyonu içermektedir. İş istasyonu, robot kolunun çalışma alanı göz önünde bulundurularak tasarlanmış bir platformdan oluşmaktadır. Sistemin içerdiği tüm elemanları bünyesinde toplamaktadır ve üzerinde çalışılacak parçaların fikstürle sabitlenmesine imkan tanımaktadır. İş istasyonu, Şekil 3.10'da gösterilmektedir.



Şekil 3.10: İş İstasyonu

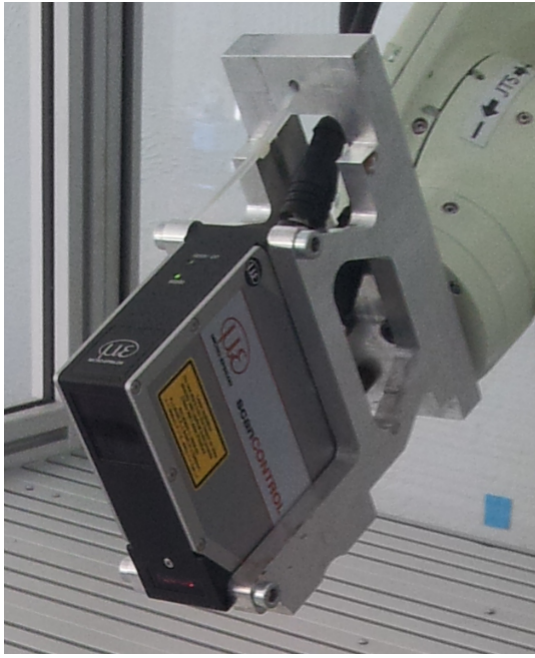
### 3.5. Güç Kaynağı

Sistemde yer alan elemanların tümüne enerji, güç kaynağı üzerinden sağlanmaktadır. Özellikle endüstriyel robot kolu için elektrik kesintileri ve tolerans dışı yüksek veya düşük gerilim gelmesi ciddi problem oluşturabilmektedir. Bu tür durumları önlemek amacıyla oluşturulan sistemde on-line teknolojiye sahip Inform DSPMP-1106 model kesintisiz güç kaynağı kullanılmaktadır.

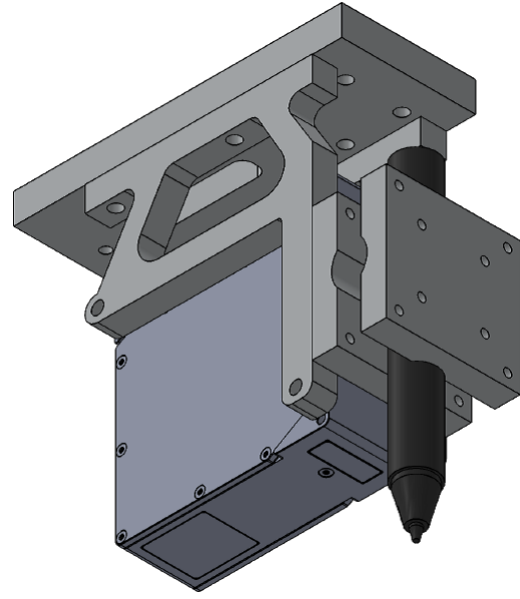
### 3.6. Algılayıcı Montaj Aparatı

Oluşturulan sistemde yer alan elemanlardan biri de algılayıcı montaj aparatıdır. Algılayıcı montaj aparatı, algılayıcının robot kol ucuna monte edilmesini sağlamaktadır. Aynı zamanda robot kol çalışma alanında bulunabilecek dış etkenlere karşı algılayıcıyı koruyacak niteliktedir ancak alüminyumdan oluşturulmuş bu parça robotun eklem hareket limitlerinin kısıtlanmasına yol açmaktadır. Algılayıcı montaj aparatının robot ucuna sensörle birlikte monte edilmiş hali, Şekil 3.11’de gösterilmektedir.

Algılayıcı montaj aparatı ile algılayıcının endüstriyel robot kolu ucuna montajı gerçekleştirilmektedir. Oluşturulan sistem gereğince yapılan ölçümlerde artık yeni uç elemanı ucu robot kolu ucu yerine algılayıcının ölçüm yapan ucu olmaktadır. Bu da algılayıcı montaj aparatı ve algılayıcının boyutuyla birlikte yeni uç elemanı ucu için z ekseninde +136 mm’lik kaymaya sebep olmaktadır. Bu değişim, yazılımsal olarak robot denetleyicisi üzerinde tanımlanmaktadır.



(a)



(b)

Şekil 3.11: Algılayıcı Montaj Aparatı (a) Görünüş (b) Teknik Çizim

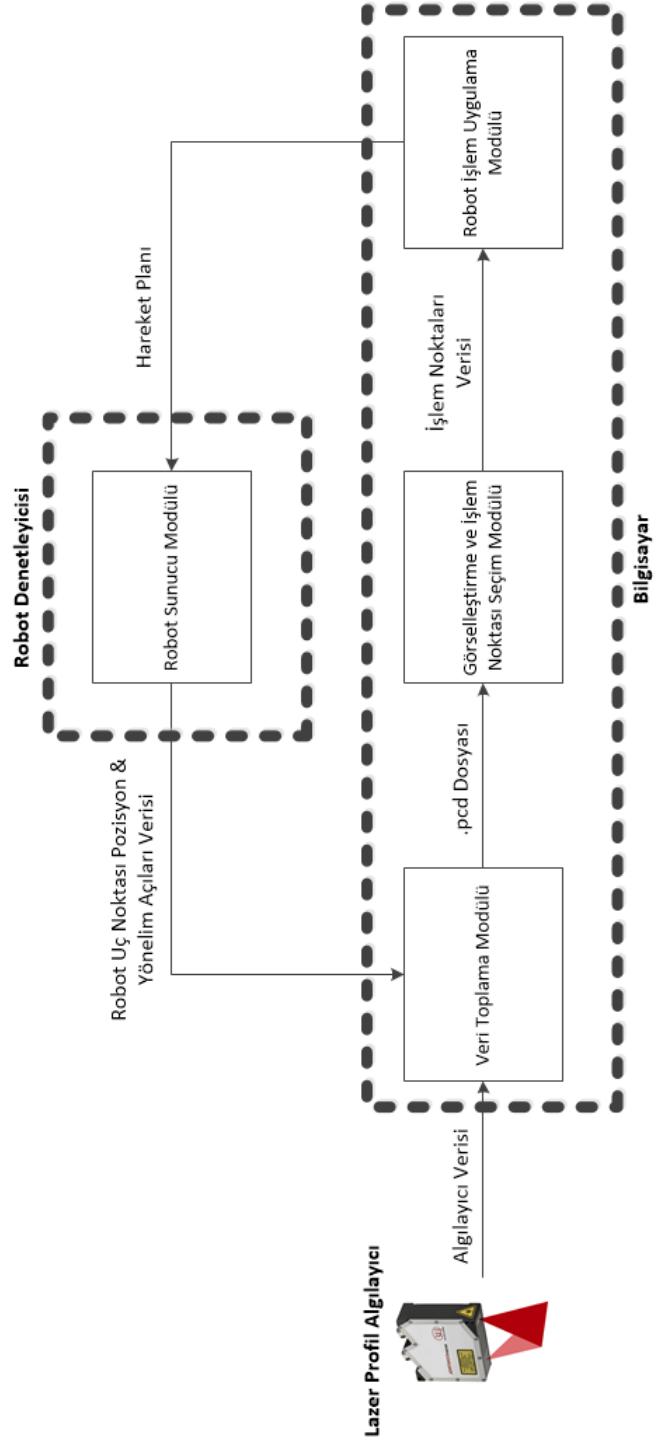
## 4. SİSTEM YAZILIMI

Geliştirilen sistem, hedeflenen amaca ulaşabilmek için yazılımsal olarak modüler yapılardan oluşmaktadır. Oluşturulan sistem yazılımı, her biri farklı amaca yönelik olacak şekilde dört farklı modülden oluşmaktadır. Bunlar robot sunucu modülü, veri toplama modülü, görselleştirme ve işlem noktası seçme modülü ile robot işlem uygulama modülleridir. Bu dört ana modül kendi içlerinde de işlevlerine göre farklı modüller içermektedir.

Veri toplama modülü olarak oluşturulan yazılımlar görselleştirme için gerekli verileri robot sunucu modülü yardımıyla toplarken, görselleştirme ve işlem noktası seçme modülü olarak oluşturulan yazılımlar ise toplanan verileri kullanarak görselleştirme işlemini gerçekleştirmenin yanısıra kullanıcıya endüstriyel parça üzerinde işlem yapılacak noktaları seçme imkanı tanımaktadır. Görselleştirme ve işlem noktası seçme modülü kullanılarak üzerinde işlem yapılacak noktaların belirlenmesiyle beraber, robot işlem uygulama modülünü oluşturan yazılımlar robotun gerekli işlemi gerçekleştirebilmesi için robota gerekli verileri yollamaktadır. Böylece kullanıcının belirlediği noktalar üzerinde gerekli endüstriyel işlem, robot sunucu modülünün yardımıyla otonom olarak yapılmaktadır. Bu akışa ait şematik Şekil 4.1’de gösterilmektedir.

### 4.1. Robot Sunucu Modülü

Robot sunucu modülü, robot üzerinde modüler yapı halinde oluşturulmuş yazılımlar bütününden oluşmaktadır. İçerdiği yazılımlar sayesinde robottan iş istasyonuna veri aktarımı, iş istasyonundan robota veri aktarımı, robotun hareketi, robotun lazer profil algılayıcıya tetikleme sinyali göndermesi gibi işlemler gerçekleştirilmektedir. Veri toplama amaçlı sunucu yazılımı ve işlem uygulama amaçlı sunucu yazılımı robot sunucu modülünü oluşturan yazılımlardır.



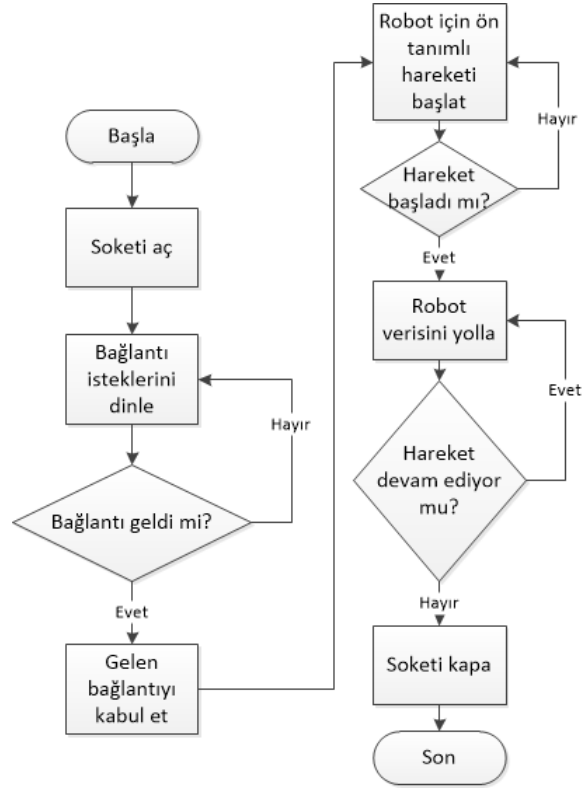
Şekil 4.1: Sistem Yazılım Tasarımı

#### **4.1.1. Veri toplama amaçlı sunucu yazılımı**

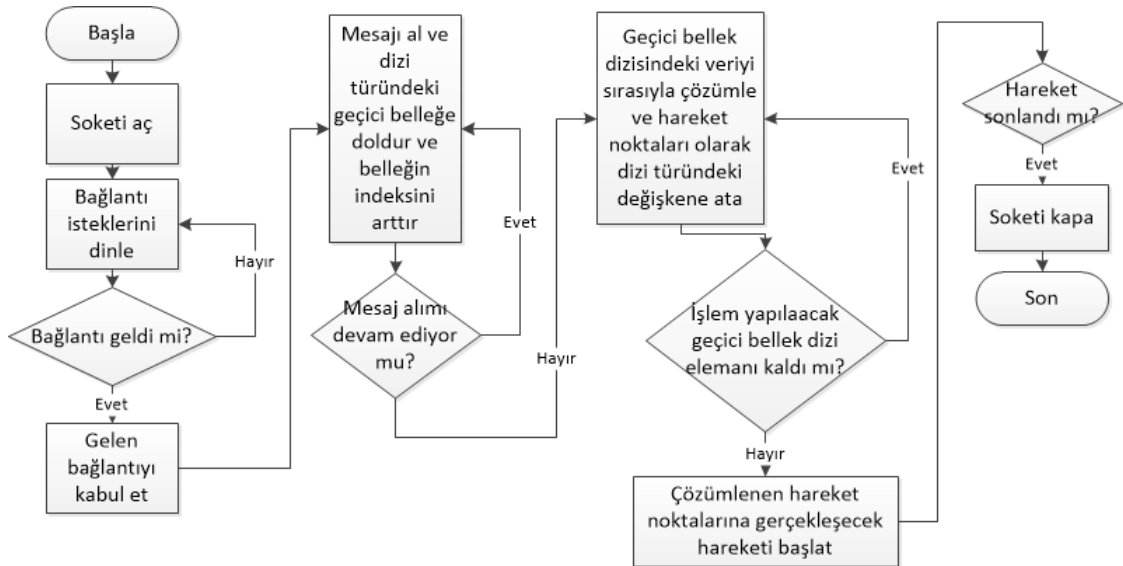
Görselleştirme işleminin bir parçası olarak robot sunucu modülü, robottan iş istasyonuna veri aktarımını gerçekleştirmektedir. Robotun üretici firması tarafından özel olarak geliştirilen AS programlama diliyle modüler yapıda oluşturulan yazılımlar bu işlemi gerçekleştirmektedir. Bu yazılımlar, veri toplama amaçlı sunucu yazılımı adı altında toplanmaktadır. Soket programlamanın kullanıldığı yazılımlarda, soketin açılması ve kapanması gibi temel yazılımların yanısıra verinin aktarımını sağlayan yazılımlar bulunmaktadır. Veri aktarım programıyla eş zamanlı olarak çalışan bir başka program da lazer profil algılayıcıya dış tetikleme sinyali yollayarak robot uç noktası ve algılayıcı ölçümlerinin senkron bir şekilde gerçekleştirilmesini sağlamaktadır. Ana program ise bu işleyişi kontrol etmektedir. Bu yazılıma ait akış diyagramı Şekil 4.2’te gösterilmektedir.

#### **4.1.2. İşlem uygulama amaçlı sunucu yazılımı**

Robot sunucu modülünün bir diğer parçasını da istenen işlemin gerçekleştirilmesine yönelik geliştirilen yazılımlar oluşturmaktadır. İşlem bilgisayarında robot işlem uygulama modülünün oluşturduğu hareket planını alarak, robot kolunun bu hareketi gerçekleştirmesini sağlamaktadır. Robot işlem uygulama modülünün çıktısı olan ve formatlı şekilde oluşturulmuş dosyayı alarak çözümlene işlemi gerçekleştiren yazılımın yanısıra, çözümlendiği dosyadaki verilere göre robotun hareketini gerçekleştiren yazılımları içermektedir. Bu işlemleri mümkün kılacak bağlantı için yine soket programlamaya başvurulmaktadır ve yine soketin açılması ve kapanması gibi temel yazılımların yanında oluşturulan ana program işleyişi kontrol etmektedir. Bu yazılıma ait akış diyagramı Şekil 4.3’te gösterilmektedir.



Şekil 4.2: Veri Toplama Amaçlı Sunucu Yazılımı Akış Diyagramı



Şekil 4.3: İşlem Uygulama Amaçlı Sunucu Yazılımı Akış Diyagramı



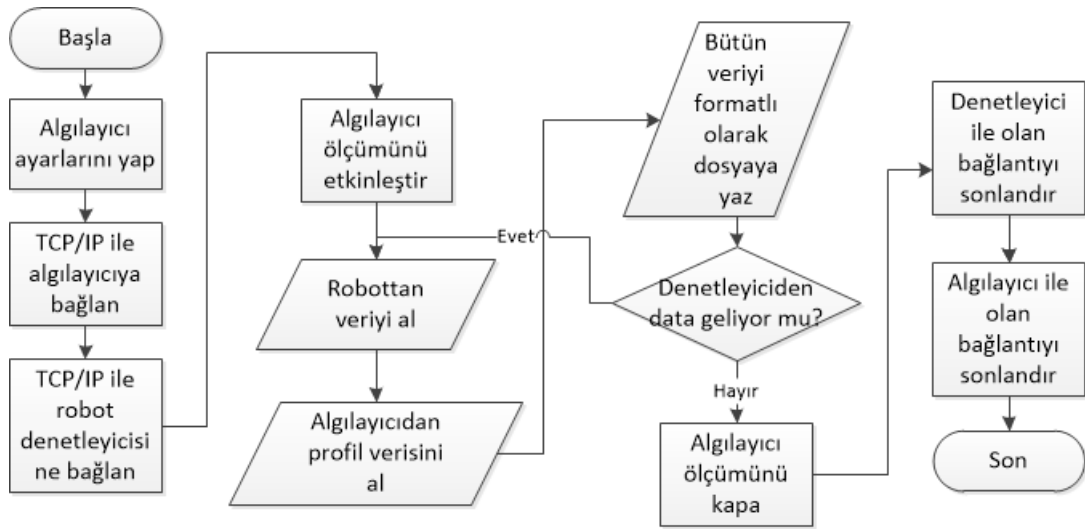
## 4.2. Veri Toplama Modülü

Veri toplama modülü, robot denetleyicisinden robot uç noktası pozisyon değerleri ve yönelim açıları ile lazer profil algılayıcıdan algılayıcı verisini toplayıp, elde edilen verilerin robot taban koordinat sistemine göre ifadesi için dönüşüm yapılmasına yarayan yazılımlardan oluşmaktadır. Bu modül ile hem robot deneyicisi, hem de lazer profil algılayıcı arasındaki veri iletimi, TCP/IP bağlantısı sayesinde gerçekleştirilmektedir ve bu yazılım socket programlama içermektedir.

Bu modül, veri toplama ve sonrasında robot taban koordinat sistemine göre ifade için dönüşüm işlemini gerçekleştiren birer adet yazılım içermektedir. Veri toplayan yazılım da istemci olarak robot üzerinde yer alan robot sunucu modülü ile iletişim sağlamaktadır. Sağladığı iletişim sayesinde robot sunucu modülünün içerdiği veri toplama amaçlı sunucu yazılımı ile birlikte çalışarak robottan veri almayı mümkün kılmaktadır. Daha sonra elde edilen veriler, dönüşüm işlemi için oluşturulan yazılımla robot taban koordinat sistemine göre ifade olacak biçime dönüştürülmektedir.

### 4.2.1. Veri toplama işlemleri

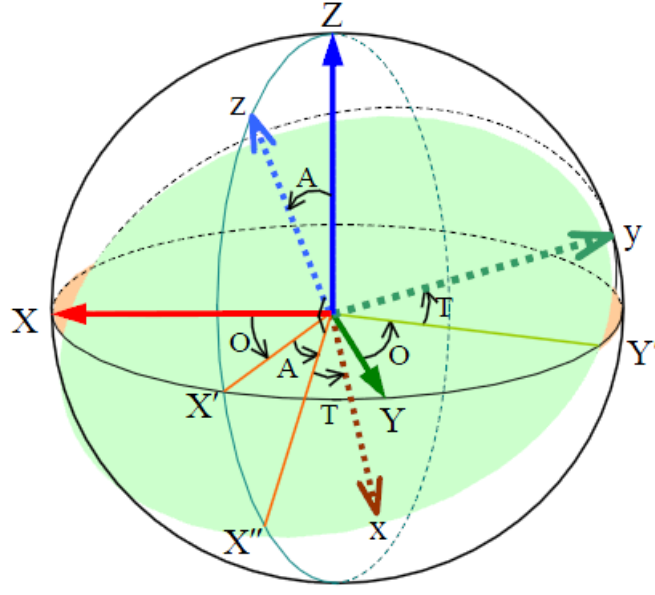
Robottan alınan veriler, robot uç noktası pozisyon değerlerini ve yönelim açılarını içermektedir. Elde edilen veriler oluşturulan yazılım aracılığıyla bir dosyaya kaydedilmektedir. Bu veriler, daha sonra görselleştirme aşamasında kullanılmaya hazır hale getirilecek şekilde algılayıcı modülünün de sağladığı verilerle birlikte dönüşüm işlemine sokulmaktadır. Bu modül aynı zamanda algılayıcıdan da algılayıcı verisini almakta ve bunun yanında sensör ölçümü ile ilgili kullanıcının istediği ayarları da yapmaktadır. Tetikleme türünün belirlenmesi, bir profilin içereceği nokta ölçüm sayısı bu ayarlardandır. Algılayıcı, her bir uç noktası pozisyon ve yönelim açısı ölçümünde robot tarafından gönderilen dış tetikleme sinyali sayesinde bu modüle algılayıcı verisini göndermektedir. Modülün bu parçasıyla elde edilen verilerin tamamı daha sonra dönüşüm işlemi için oluşturulan yazılım tarafından kullanılmaktadır. Bu modülün gerçekleştirdiği işlemlerin dönüşüm işlemine kadar olan akışını gösteren akış diyagramı Şekil 4.4'te gösterilmektedir.



Şekil 4.4: Veri Toplama Modülünün Veri Toplama İşlemi Akış Diyagramı

#### 4.2.2. Dönüşüm işlemleri

Oluşturulan sistemde endüstriyel robot kolundan, robot uç noktasının pozisyon değerlerine ve yönelim açılarına dair okunan veriler robot taban koordinat sistemine göre ifade edilmektedir. Robot denetleyicisi, robot uç noktası koordinat sistemi yönelimini robot taban koordinat sistemine göre ifade ederken Euler açılarını kullanmaktadır. Şekil 4.5’de gösterildiği üzere üç Euler açısı O, A ve T açıları şeklinde tanımlanmaktadır. Taban koordinat sistemi X, Y ve Z eksenlerinden oluşurken, hedef koordinat sistemi x,y ve z eksenlerinden oluşmaktadır. Bu duruma göre O açısı taban koordinat sisteminin, Z eksenini etrafında rotasyonunu tanımlamaktadır. Bu rotasyon sonrasında oluşan yeni koordinat sistemi X', Y' ve Z eksenlerine sahip olmaktadır. Yeni koordinat sisteminin Y' eksenini etrafında rotasyonu da A açısını tanımlamaktadır. Bu rotasyon sonrasında da yeni koordinat sistemi X'', Y' ve z eksenlerini içermektedir. Bu koordinat sisteminin, z eksenini etrafında rotasyonu da son olarak T açısını tanımlamaktadır. Bu açılar, sonuç olarak algılayıcıdan elde edilen verilerin robot taban koordinat sistemine göre ifade edilebilmesi için dönüşüm işlemlerinde kullanılmaktadır.



Şekil 4.5: Robot Denetleyicisinde Euler Açılı İfadeleri (Kawasaki Heavy Industries, 2015)

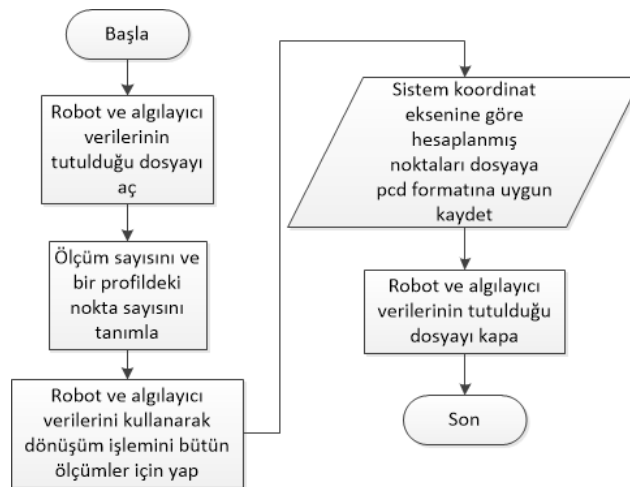
Lazer profil algılayıcıdan elde edilen ölçümler, herhangi bir dönüşüm gerçekleştirilmediğinde lazer profil algılayıcı uç noktasına göre pozisyon bilgisi içermektedirler. Dolayısıyla bu ölçümlerin görselleştirme işlemi yapılmadan önce robot taban koordinat sistemine göre ifade edilmesi gerekmektedir. Bunu sağlamak amacıyla Denklem 1’de verilen D dönüşüm matrisi kullanılarak dönüşüm işlemi gerçekleştirilmektedir. Bu eşitlikte R matrisi, 3x3 boyutlarında ve Euler açıları Z-Y-Z konvansiyonuna göre düzenlenmiş rotasyon matrisidir ve robot denetleyicisinin sunduğu yönelim açılarını kullanmaktadır. R matrisi Denklem 2’de gösterilmektedir. P matrisi ise Denklem 3’de gösterildiği gibi uç noktasının pozisyonunun robot taban koordinat sistemine göre ifadesini içermektedir ve robot denetleyicisinin sunduğu uç konum koordinatlarını kullanmaktadır.

$$D = \begin{bmatrix} R_{3x3} & P_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} \quad (1)$$

$$R = \begin{bmatrix} \cos \varphi \cos \theta \cos \psi - \sin \varphi \sin \psi & -\cos \varphi \cos \theta \sin \psi - \sin \varphi \cos \psi & \cos \varphi \sin \theta \\ \sin \varphi \cos \theta \cos \psi + \cos \varphi \sin \psi & -\sin \varphi \cos \theta \sin \psi - \cos \varphi \cos \psi & \sin \varphi \sin \theta \\ -\sin \theta \cos \psi & \sin \varphi \cos \psi & \cos \theta \end{bmatrix} \quad (2)$$

$$P = \begin{bmatrix} X_{unp} \\ Y_{unp} \\ Z_{unp} \end{bmatrix} \quad (3)$$

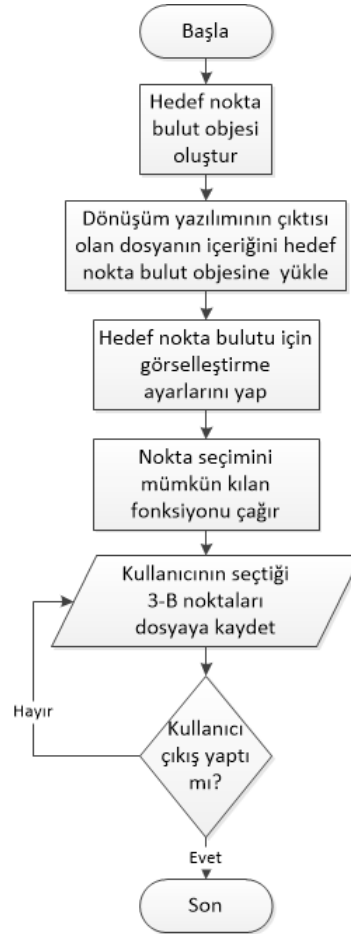
Dönüşüm işlemini gerçekleştiren yazılım, robot uç noktası pozisyon ve yönelim açı değerlerinin yanında algılayıcıdan elde ettiği verileri de kullanıp, algılayıcı uç noktasına göre elde edilen ölçümlerin robot taban koordinat eksenine göre ifade edilmesini sağlamaktadır. Çıktı olarak ise PCL platformuna uygun nokta bulut veri dosyası olacak şekilde ".pcd" uzantılı dosya üretmektedir. Bu dosya, daha sonra görselleştirme ve işlem noktası seçme modülü tarafından görselin oluşturulmasında kullanılmaktadır. Dönüşüm yazılımına ait akış diyagramı Şekil 4.6'de gösterilmektedir.



Şekil 4.6: Veri Toplama Modülünün Dönüşüm İşlemi Akış Diyagramı

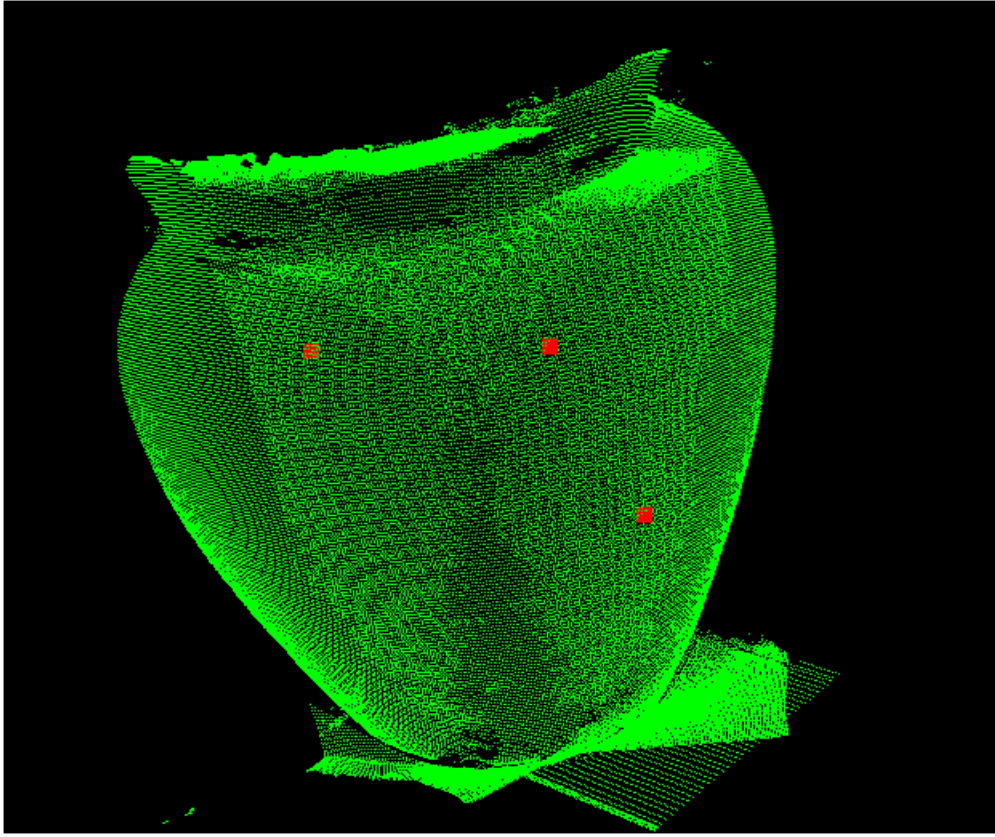
### 4.3. Görselleştirme ve İşlem Noktası Seçim Modülü

Görselleştirme ve işlem noktası seçme modülünün içerdiği yazılım, PCL adlı açık kaynak kodlu kütüphanenin kullanımını içermektedir. Bu yazılımın görevi, veri toplama modülünün içerdiği yazılımlar tarafından oluşturulan nokta bulut verisini kullanıp görselleştirme işlemini gerçekleştirmenin yanısıra, kullanıcıya görsel üzerinden istenilen 3-boyutlu noktaya ait bilgileri öğrenme fırsatı da sunmaktır. Öyle ki kullanıcı görsel üzerinden klavye ve fare gibi girdi araçları sayesinde bir nokta seçimi yaptığında, bu noktaya ait pozisyon bilgileri bir dosyaya kaydedilmektedir. Görselleştirme ve işlem noktası seçim yazılımına ait akış diyagramı Şekil 4.7’de gösterilmektedir.



Şekil 4.7: Görselleştirme ve İşlem Noktası Seçim Modülü

Görselleştirme ve işlem noktası seçme modülü sonuç olarak elde edilen 3-boyutlu nokta bulut verisinin görselleştirilmesinin yanında kullanıcıya endüstriyel parça üzerinde işlem gerçekleştirilecek noktaları belirlemesine de imkan tanımaktadır. Bir vazoya ait örnek bir nokta bulut görselleştirmesi üzerinde yapılan ve seçilen üç adet noktanın farklı bir renk ile işaretlendiği işlem noktası seçimi sonucu oluşan görüntü Şekil 4.8’de gösterilmektedir.



Şekil 4.8: Görselleştirme Üzerinde İşlem Noktası Seçim Örneği

#### 4.4. Robot İşlem Uygulama Modülü

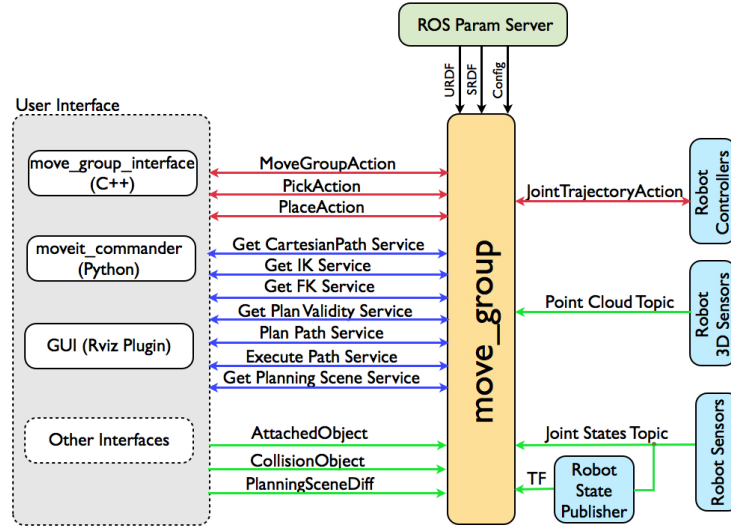
Robot işlem uygulama modülü, kullanıcın görselleştirme modülü aracılığıyla belirlediği noktaları işleyip robot için hareket planı oluşturan ve bunu robota ileten yazılımlardan oluşmaktadır. Belirlenen noktalar için engellerden sakınan hareket planı oluşturarak bu hare-

ket planını robota ileten bu modül, hareket planını simüle ederek oluşturan ve oluşan hareket planında belirli anlardaki eklem açı değerlerini robota ileten yazılımları içermektedir.

Bu tez çalışmasında hareket planlama aşaması, Linux tabanlı bir işletim sistemi olan Ubuntu üzerinde Open Motion Planning Library (OMPL) isimli açık kaynak kodlu hareket planlama kütüphanesinin ve MoveIt isimli yazılımın kullanımını içermektedir.

OMPL isimli açık kaynak kodlu hareket planlama kütüphanesi, birçok güncel örnekleme tabanlı hareket planlama algoritması içermektedir. PRM, RRT, EST, SBL, KPIECE, SyCLOP gibi güncel örnekleme tabanlı algoritmaları ve bu planlayıcıların türevleri, kütüphanenin içerdiği örnekleme tabanlı algoritmalarındandır. Bütün planlayıcılar soyut olarak tanımlanmış durum uzaylarında çalışmaktadır ve genel olarak başvurulmuş birçok durum uzayı kütüphanede tanımlanmış durumdadır. Tek başına OMPL kütüphanesi, çarpışma kontrolü ve görselleştirme işlemleriyle alakalı herhangi bir işlem yürütmemektedir. Bu işlemler, OMPL için geliştirilen OMPL.app isimli uygulama veya MoveIt gibi OMPL kütüphanesiyle birlikte çalışabilen yazılımlar aracılığıyla gerçekleştirilebilmektedir.

Bu çalışmada, engellerden sakınma ve oluşan hareket planının görselleştirilmesi amaçlarından yola çıkarak OMPL kullanımını sağlayan MoveIt yazılımı kullanılmaktadır. MoveIt yazılımı, mobil manipülasyon, manipülasyon, 3-boyutlu algılama, kinematik, kontrol ve navigasyon alanlarının yanında hareket planlama alanındaki son geliştirmeleri bünyesinde toplayan bir yazılımdır. Açık kaynak kodlu ve robotlar için işletim sistemi olarak tanımlanan "Robot Operating System" (ROS) bünyesinde çalışan bu yazılım, hareket planlama, planlama ortamının oluşturulması, kinematik, çarpışma kontrolü ve yörünge işleme gibi süreçleri yürütebilmektedir. Bütün bu süreçler, Şekil 4.9'da gösterildiği gibi bir sistem mimarisine sahip olan "movegroup" isimli ana düğüm aracılığıyla yürütülmektedir. MoveIt yazılımı, bu ana düğüm ve ROS platformunun sunduğu imkanlarla beraber bir robot modelini kullanarak hareket planlama işleminin gerçekleştirilmesine imkan tanımaktadır.



Şekil 4.9: MoveIt Yazılımının İçerdiği Ana Düğüm için Üst Seviye Sistem Mimarisi (Sucan and Chitta, 2013)

MoveIt yazılımı, hareket planlayıcıları eklentiler aracılığıyla kullanmaktadır. MoveIt yazılımının sunduğu varsayılan hareket planlayıcı, OMPL kütüphanesi kullanımıyla yapılandırılmaktadır. 3-boyutlu algılama işlemi de yine eklentiler aracılığıyla sağlanmaktadır. Nokta bulut verisi gibi birçok türden algılayıcı verisi, 3-boyutlu algılama için kullanılabilir. MoveIt yazılımı daha sonra, ortamın yertutum haritasını algılayıcı verisini kullanarak OctoMap kütüphanesi aracılığıyla sağlamaktadır. OctoMap kütüphanesi, 3-boyutlu yertutum örgü haritalama yaklaşımını uygulayan bir kütüphanedir ve haritalama uygulaması "octree" anlayışına dayanmaktadır.

MoveIt yazılımı ve içerdiği OMPL kullanımı sayesinde oluşturulan hareket planı, daha sonra robot işlem uygulama modülü aracılığıyla robottaki işlem uygulama amaçlı sunucu yazılıma iletilmektedir. Bu işlem için ilk olarak oluşan hareket planında takip edilen yörünge süresince eklem açısı değerlerinin aldığı değerler ROS üzerinde bir istemci yazılım aracılığıyla belirli aralıklarla toplanıp bir dosyaya yazılmaktadır. Belirli zaman aralıklarıyla toplanan eklem açısı değerleri daha sonra robotun oluşan hareketi gerçekleştirebilmesi için ethernet bağlantısı aracılığıyla robottaki işlem uygulama amaçlı sunucu yazılıma gönderilmektedir.



#### 4.4.1. Hareket planlama

Robotlar için hareket planlaması, robotun başlangıç durumundan hedef durumuna ulaştırılması problemine robotun ve içinde bulunduğu ortamın getirdiği kısıtlar da göz önünde bulundurularak çözüm aranmasıdır. Bu çözüm arayışı, hesaplama açısından bakıldığında robotun sahip olduğu eksen sayısı arttıkça zorlaşmaktadır. 6-eksenli bir robot kolunun sahip olduğu serbestlik derecelerinden üçü koordinat düzlemlerinde hareket için, üçü de koordinat düzlem eksenleri etrafındaki yönelimi temsil etmek için kullanılmaktadır. Böyle bir robot kolunu başlangıç konfigürasyonundan hedef konfigürasyonuna engellerden sakınarak ulaştıran kesin çözümü hesaplamak oldukça zordur. Bu işlem, altı serbestlik derecesinde olan devamlı değişimleri hesaplamakla mümkün olmaktadır. Dolayısıyla hareket planlama problemlerinde bu zorluğun üstesinden gelmek için örnekleme tabanlı hareket planlama yaklaşımı gibi çeşitli yaklaşımlar kullanılmaktadır.

Robotik alanında robotların hareket planlaması için literatürde birçok farklı yaklaşım bulunmaktadır. Kesin ve yaklaşık hücre ayrıştırması, kontrol tabanlı metodlar, potansiyel alanlar, rastgele hale getirilmiş planlama ve örnekleme tabanlı planlama bu yaklaşımlar arasında yer almaktadır.

Kesin ve yaklaşık hücre ayrıştırması, ortamın engel içermeyen kısımlarını ayrık hücrelere ayırarak çalışma alanını parçalara ayırma esasına dayanmaktadır. Bu ayrıştırma, daha sonra hücrelerin sahip olduğu iki nokta arasında bir yörünge oluşturmayı kolaylaştırmaktadır. Hücre ayrıştırma yaklaşımı, konfigürasyon uzayının parçalara ayrıldığı kesin hücre ayrıştırması ve sıradan hücre şekilleriyle konfigürasyon uzayının yaklaşık ifadesinin oluşturulduğu yaklaşık hücre ayrıştırması şeklinde iki yaklaşıma sahiptir.

Chazelle (1987) tarafından yapılan çalışmada ortaya koyulduğu gibi kesin hücre ayrıştırması yaklaşımında, poligonal engellere sahip 2-boyutlu bir ortamda basitlik ve uygunluk sağlayan bir metod olduğu için genellikle dikey ayrıştırma metodu uygulanmaktadır. Bu metodda hücreler ve hücre bağlantı grafiği, bir koordinat ekseninde bir tarama çizgisinin kaydırılarak bu çizgiyle kesişen hücrelerin ve kenarların kaydının tutulmasıyla

oluşturulmaktadır. Hücreler üçgensel veya ikizkenar yamuk şeklinde olup, iki engelin kenarlarıyla sınırlandırılmakta ve bir veya iki kenar tarama çizgisine paralel olmaktadır. Yeni bir engelin tepe noktası ile karşılaşıldığında hücreler kümesi, o tepe noktasına temas eden hücre elenecek şekilde güncellenmektedir ve tepe noktasından uzanan kenarın yönelimine bağlı olarak bir veya iki hücre tarama çizgisinin devamına eklenmektedir. Böylece yeni eklenen hücreler, hücre bağlantı grafiğine eklenerek elenen hücrelere bağlanmaktadır. Hücre ayrıştırması sonlandığında ise yörünge planlama sorgulaması da başlangıç ve hedef noktalarını içeren hücreler arasında grafik araması ile gerçekleştirilmektedir.

Brooks ve Perez (1983) tarafından ortaya koyulduğu gibi yaklaşık hücre ayrıştırması da kesin hücre ayrıştırmasından farklı olarak genellikle kare veya hiper küp şeklindeki düzenli şekle sahip hücreleri kullanarak konfigürasyon uzayını düzenli parçalara ayırmaktadır. Bu hücreler daha sonra kısmen engel içermesi, tamamen engel içermesi veya içermemesi durumuna göre karma, dolu veya boş olarak etiketlenmektedir. Hücrelerin oluşturulması bu şekilde yinelemeli olarak ve hücre ağacı oluşturacak şekilde gerçekleştirilmektedir. 2-boyut için kare hücrelerin genişlemesi her bir yönde olmakta, dolayısıyla oluşan hücre ağacı da dördün ağaç şeklinde oluşmaktadır. 3-boyut için ise oluşan arama ağacı sekizlik ağaç olmaktadır. Kesin hücre ayrıştırması yaklaşımında olduğu gibi yaklaşık hücre ayrıştırmasında da başlangıç ve hedef noktalarını içeren hücreler belirlendikten sonra, bu hücreler arasındaki bağlantı grafiği grafik aramasıyla taranarak sadece karma ve boş hücreleri içerecek şekilde bir yörünge bulunmaktadır. Oluşan yörünge karma bir hücre içeriyorsa, bu hücreler yine ayrıştırılarak yeni bir grafik araması gerçekleştirilmektedir ve bu süreç tamamen boş hücreler içeren bir yörünge bulunana kadar devam etmektedir.

Kesin ve yaklaşık hücre ayrıştırması tümyönlü tabana sahip bir robot için elverişli bir yaklaşım olsa da, kompleks ve lineer olmayan dinamiklere sahip robotlar için robotun bir hücreden diğerine hareketi net olmayacağından elverişsizdir. Kesin hücre yaklaşımı için ise belirgin poligonal bir konfigürasyon uzayı mevcut değilse uygulanması kullanışsız olmaktadır. Aynı şekilde van den Berg ve Overmars (2005) tarafından hibrit bir tekniğin sunulduğu çalışmada belirtildiği üzere robot hareketleri yüksek boyutlardaki konfigürasyon uzaylarında hesaplandığı için bu uzayları hücrelere ayırmak hücre ayrıştırma metodlarını ek bir tekniğin varlığı olmaksızın elverişsiz kılmaktadır.

Kontrol tabanlı metodlar ise sistemin hareket denklemlerini modelleyerek kontrol te-ori yaklaşımlarını bu modellemelere uygulamaya yönelik yaklaşımları içermektedir. Bu yaklaşımlar devamlı uzaylara uygulanmaktadır ve sistemin manevralarında minimal hata elde edebilmek için geri besleme döngüsü kullanılmaktadır. Kontrol tabanlı yaklaşımlar ile hareket planlaması oldukça hızlı ve çevrimiçi şekilde oluşturulabilmektedir ancak kompleks dinamiklere sahip sistemler için istenen ve uygulanabilir bir yörüngeyi elde etmek zor olabilmektedir.

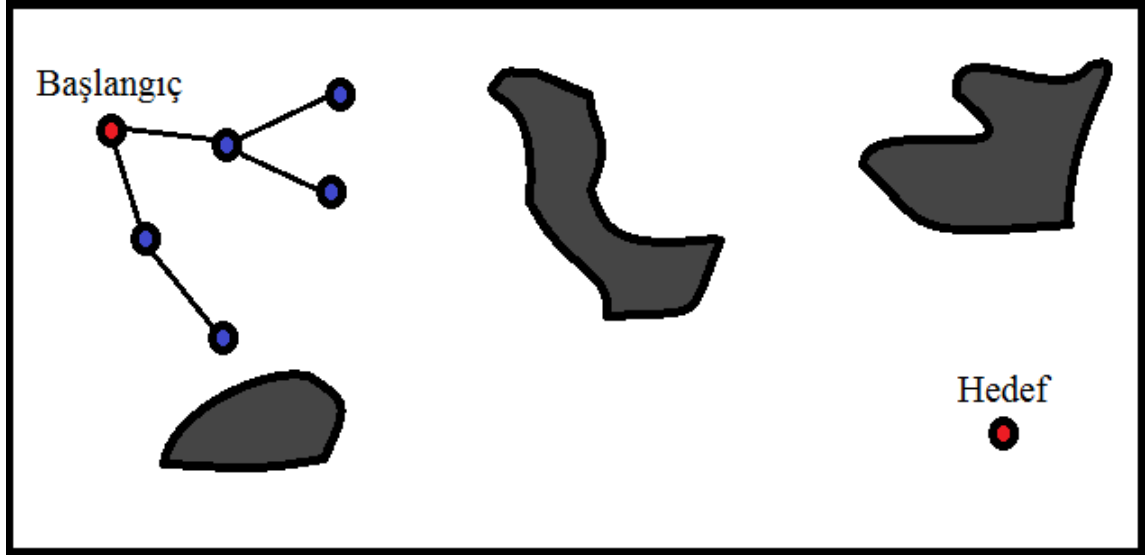
Potansiyel alanların hareket planlaması için kullanımı ise basit ve sezgisel bir yaklaşımdır. Khatib (1986) ortaya koyduğu çalışmasında ilk olarak potansiyel alanları robot kollarının hareket planlaması için kullanmıştır. Klasik potansiyel alan yaklaşımı, hedefin çekici güç, engellerin ise itici güç teşkil ettiği çalışma alanındaki her bir noktada vektörlerin toplanması hesabını içermektedir. Nokta robotu, daha sonra gradyan iniş metoduyla potansiyelleri takip ederek hedefe ulaşmaktadır. Bu yaklaşımda hareketin yönüne karar vermek için lokal bilgi kullanıldığından robotun lokal minimalarda çıkmaza girip hareket planlamayı başarısızlığa uğratması da söz konusu olabilmektedir. Potansiyel alanlar, robot üzerinde çoklu noktalar düşünülerek katı cisimler için genelleştirilebilmektedir ve gerçek zamanlı çalışabilmektedir.

Örnekleme tabanlı hareket planlama ise robotun durum uzayının örneklemesini kullanarak özellikle difransiyel kısıtlara ve çok sayıda serbestlik derecesine sahip olan sistemlerin planlama sorgularına hızlı ve efektif bir şekilde cevap vermektedir. Geleneksel yaklaşımların kullanımıyla bu tarz sistemlerin problemlerinin çözümü, hareket kısıtları ve durum uzayının boyutu sebebiyle uzun sürebilmektedir. Dolayısıyla örnekleme yaklaşımı, büyük ve karmaşık yapıdaki durum uzayını kapsayarak başlangıç ve hedef noktaları arasında uygulanabilir bir yörünge ortaya koyma ihtiyacından ortaya çıkmaktadır.

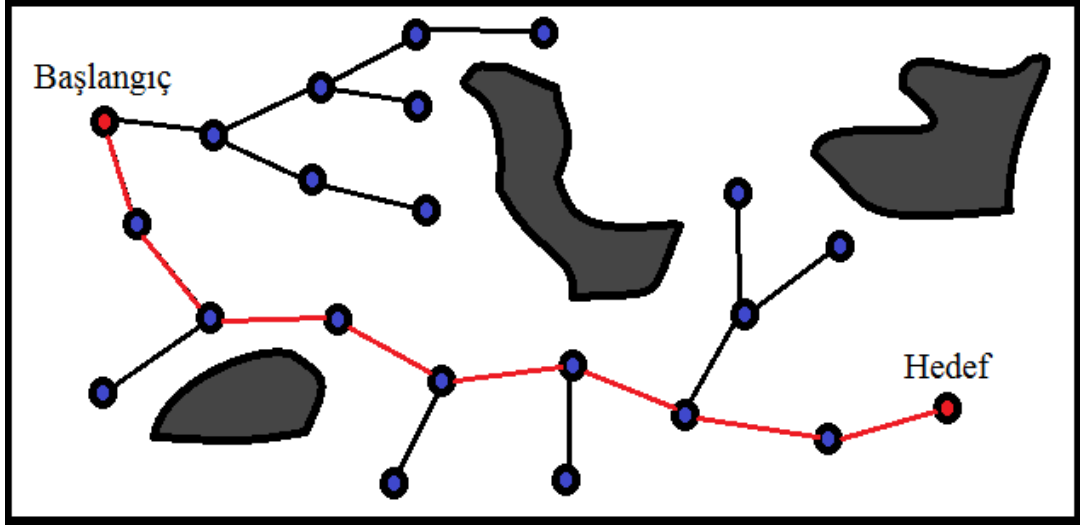
Örnekleme tabanlı hareket planlayıcılar, takip ettikleri metod açısından olasılıksal yol haritası ve ağaç tabanlı planlayıcılar olmak üzere sınıflandırılabilir. Kavraki vd. (1996), çalışmalarında sundukları olasılıksal yol haritası tabanlı metodla örnekleme tabanlı hareket planlayıcıların ilk örneklerini vermişlerdir. Bu yaklaşım, serbest durum uzayında bir yol haritası oluşturmak için durum uzayının rastlantısal örneklendirilmesini kullanmaktadır.

Örnekleme tabanlı planlayıcıların büyük bir kısmı ise serbest durum uzayında ağaç yapısı kullanmaktadır. Yaratılan ağaç yapısı olasılıksal yol haritası metoduyla benzerlik taşımaktadır. Bu metodu içeren planlayıcılar, robotun başlangıç konfigürasyonundan itibaren başlayarak kökler oluşturmaktadır. Robot, bu ağaç üzerinde bir nokta ile ifade edilmektedir ve ağaç üzerinde ilk düğümle beraber boş alanın rastlantısal örnekleme başlanmaktadır. Planlayıcı, keşifsel ilerlemeyle daha sonra çarpışmasız bir yörünge için örnekler oluşturmaktadır. 2-boyutlu bir çalışma alanında ağaç üzerinde oluşmuş uygun örneklerin gösterildiği örnek bir gösterim Şekil 4.10'de gösterilmektedir.

Örnekleme aşamasında oluşturulan bir örneğin tam olarak hedef noktası olması düşük olasılıklı bir durum olduğu için, örnekleme tabanlı planlayıcılar genellikle ağacın genişlemesini hedef duruma yanlılık olacak şekilde oluşturma amacındadır. Hedef durumu çözüm ağacına bağlayabilmek mümkün olduğunda ise ilgili çözüm çevriminde hedef durumu ağaca bağlanarak arama sonlandırılmaktadır. Hedef durumun ağaca bağlanarak aramanın sonlandırılmasına ilişkin gösterim Şekil 4.11'de gösterilmektedir.



Şekil 4.10: Örnekleme Aşamasında Ağaç Üzerinde Örneklerin Oluşması



Şekil 4.11: Örnekleme Aşamasında Hedef Durumun Ağaca Bağlanarak Aramanın Sonlandırılması Durumu

Ağaç tabanlı hareket planlayıcılar ve yol haritası tabanlı hareket planlayıcılar arasındaki farklar, üzerinde çalışılacak sistem için önem taşımaktadır. LaValle (1998) yaptığı çalışmada gösterdiği üzere ağaç tabanlı planlayıcılar daha çok tekil sorgulama tabanlı planlama için uygun planlayıcılardır. Aynı zamanda sahip oldukları veri yapısından dolayı karmaşık dinamik yapısına sahip sistemlerde planlama için mevcut örnekleme planlı yöntemlere göre daha uygundurlar.

Bu tez çalışmasındaki hareket planlama sürecinde, örnekleme dayalı ağaç tabanlı planlayıcılardan biri olan Rapidly-exploring Random Trees (RRT) algoritmasının bir varyasyonu olan RRT-Connect algoritması kullanılmaktadır. Kuffner ve LaValle (2000) tarafından sunulan çalışmada ortaya koyulduğu üzere RRT-Connect algoritması, varyasyonu olduğu RRT algoritmasından farklı olarak arama ağacını başlangıç durumunun yanında hedef durumundan da başlatmaktadır ve iki ağaç birbirlerine doğru genişlemektedir. Connect sezimi (heuristic) de arama ağacının bir adım yerine çoklu adım atmasını sağlayarak arama ağacının oluşumunda daha fazla yol kat etmeyi sağlamaktadır.

RRT-Connect algoritması, ağacı genişletmeyi RRT algoritmasında olduğu gibi tek bir adım genişletmek yerine başlangıç ve hedef durumları birbiriyle buluşana veya engelle

karşılaşılanlara kadar sürdürmektedir. Potansiyel alanlar ile planlamada ortaya koyulan yapay potansiyel fonksiyonla da benzerlik gösteren bu yaklaşımda sahip olunan sezim, çözüme hızlı bir yakınsama sağlamaktadır. Bu metodla birlikte fırsatçı sezim, RRT algoritmasının hızlı ve düzenli keşfetme özellikleriyle birleşerek yerel en küçükten kaynaklanan sorunları engellemektedir. RRT-Connect algoritması Şekil 4.12’de gösterilmektedir.

---



---

### **RRT-Connect (Başlangıç, Hedef)**

Birinci Kök  $\Leftarrow$  Başlangıç

İkinci Kök  $\Leftarrow$  Hedef

**Döngüyü** 1’den iterasyon sayısına kadar başlat

Z  $\Leftarrow$  rastgele pozisyon düğümü

Y  $\Leftarrow$  Z düğümüne en yakın RRT düğümü

X  $\Leftarrow$  Y düğümünün başlangıca/hedefe uzanması

**Eğer** Y  $\Rightarrow$  X uygulanabilir ise ve Z RRT üzerinde bulunmuyorsa

X’in üst düğümü  $\Leftarrow$  Y

X düğümünü RRT ağacına ekle

**Eğer** Connect(Birinci Kök, İkinci Kök, X) geçerliyse

Birinci Kök ve İkinci Kök döndür

Birinci Kök ve İkinci Kök’ü kendi arasında değiştir

**Döngüyü** sonlandır

### **Connect (Birinci Kök, İkinci Kök, Z)**

Önce

X’i Birinci Kök’e genişlet

Başlangıca/hedefe ulaşılmamışsa veya engellerle çarpışma olmadıysa yapmaya devam et

**Eğer** başlangıca/hedefe ulaşılmışsa

Fonksiyon geçerli değer döndürsün

**Aksi takdirde**

Fonksiyon geçersiz değer döndürsün

---

Şekil 4.12: RRT-Connect Algoritması

#### 4.4.2. ROS tabanlı hareket planlama

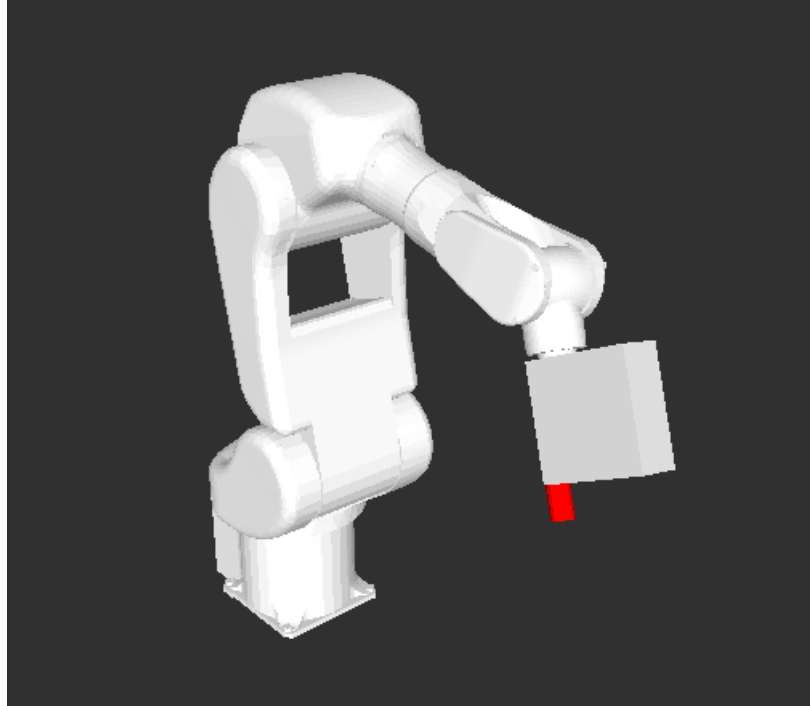
Bu tez çalışmasında hareket planlaması için kullanılan yardımcı yazılım MoveIt, bünyesinde barındırdığı robot modellerinin dışındaki bir robot modeli için konfigürasyon ayarlarına ihtiyaç duymaktadır. Bu ayarlar ise MoveIt yazılımının sunduğu konfigürasyon sihirbazı kullanılarak yapılabilmektedir. Konfigürasyon sihirbazı girdi olarak üzerinde çalışılacak robotun ROS platformuna özel bir dosya türü olan "urdf" formatında tanımlanmış modelini gerektirmektedir. Bu kapsamda, hareket planlaması üzerine yapılacak çalışmaların ilk kısmını robot modelinin oluşturulması oluşturmaktadır.

Modelleme işlemi için kullanılan "urdf" dosya formatı, aslında robot modelini tanımlamak için kullanılan bir XML formatıdır. Bu format, ROS yapısının sahip olduğu "urdf" paketindeki C++ ayrıştırıcı ile işlenebilmektedir. Bu "urdf" paketi, robot modeli, sensör, ortam vb. bileşenler için birçok XML tanımlaması içermekte ve her bir XML tanımlaması için de bir ayrıştırıcı içermektedir.

"Urdf" dosya formatı, "link", "transmission", "joint", "sensor" ve "model" gibi XML tanımlamaları içermektedir. Bu ana başlıklar altında olmak üzere, her bir XML tanımlaması yine kendi içerisinde XML tanımlamaları şeklinde detaylandırmalara sahiptir. Her bir XML tanımlaması içerdiği alt tanımlamalar sayesinde modeli detaylandırmaya imkan tanımaktadır.

Robot modeli oluşturulurken "stl" dosya tipinde oluşturulmuş link modelleri kullanılmaktadır. Sahip olunan yedi link modeli "urdf" dosyasında robot sağlayıcısının sunduğu kinematik ve dinamik özellikler dikkate alınarak birleştirilmektedir. Her bir link, kendisini tanımlayan "stl" dosyalarının "urdf" dosyasında "mesh" etiketiyle kullanılmasıyla modele eklenmektedir. Her bir link birbirine yine robot sağlayıcısının sunduğu bilgiler ışığında eklemlerle bağlanmaktadır. Eklemler için temel olarak orjin, eklem türü ve dönel olanlar için hareket limitleri tanımlanmaktadır. Bu tanımlamalar sonunda ROS platformu üzerinde, gerçekleştirilmesi istenen sürecin simülesi gibi birçok uygulamada kullanılacak bir "urdf" modeli elde edilebilmektedir. Bu çalışmada oluşturulan modele ait görsel Şekil 4.13'de gösterilmektedir. Modele ait urdf dosyasının içeriği ise Ek Açıklamalar-A'da verilmektedir.

Hareket planlamasını MoveIt aracılığıyla tanımlı olmayan bir robot modeli için gerçekleştirirken robot modelinin yanında, bu model için oluşturulacak MoveIt konfigürasyon paketine ihtiyaç duyulmaktadır. Bu sebeple model oluşturulduktan sonraki aşamayı da MoveIt konfigürasyon paketinin oluşturulması oluşturmaktadır. MoveIt yazılımı, öntanımlı olmayan robot modellerinin konfigürasyon ayarlarının yapılabilmesi için "MoveIt Setup Assistant" olarak tanımladığı bir arayüz sunmaktadır. Bu arayüz aracılığıyla ilk olarak tanımlı robot modeli için öz çarpışma matrisi üretilmektedir. Bu matris, çarpışma kontrolü yapılması gerekli olmayan robot linklerini saptamak için üretilmektedir. Bu matris üretilirken robot rastgele çok sayıda farklı pozisyona sokulmaktadır ve böylece ilgili linkler saptanmaktadır. Bu matrisin üretilmesiyle birlikte hareket planlama işlem süresi kısaltılmaktadır. Hareket planlamasının gerçekleştireceği model, arayüz üzerinde ekleleriyle birlikte planlama grubu olarak tanımlanmakta ve MoveIt yazılımının sunduğu bir eklenti kinematik çözücü olarak kullanılmaktadır. Bu işlemlerin tamamlanmasıyla birlikte MoveIt yazılımının kullanılacağı konfigürasyon dosyaları ilgili model için oluşturulmaktadır.



Şekil 4.13: Kawasaki RS05L İçin Oluşturulan Urdf Modeli



Bu tez çalışmasında robot modeli için hareket planlaması, MoveIt yazılımının hareket planlayıcı olarak sahip olduğu Open Motion Planning Library (OMPL) kütüphanesinin kullanılmasıyla gerçekleştirilen, oluşan hareket planının simüle edilmesi yine MoveIt yazılımının sahip olduğu "Rviz" isimli eklenti sayesinde sağlanmaktadır. Dolayısıyla bu tez çalışmasında OMPL kütüphanesinin içerdiği hareket planlama algoritmalarına başvurulmaktadır.

OMPL kütüphanesi, LBKPIECE, PRM, RRT, RRT-Connect gibi birçok hareket planlama algoritması içermektedir. Bu algoritmaların kullanımı ve algoritmaların sahip oldukları parametrelerin ayarlanması MoveIt yazılımının başvurduğu "ompl\_planning.yaml" adlı konfigürasyon dosyası aracılığıyla olmaktadır. Bu dosya, içerdiği tüm algoritmalar için "range", "goal\_bias" gibi çeşitli parametreler içermektedir.

Bu tez çalışmasında OMPL kütüphanesinin içerdiği RRT-Connect hareket planlama algoritması kullanılmaktadır. RRT-Connect hareket planlama algoritması için konfigürasyon dosyasında tanımlanan parametreler "range" ve "longest\_valid\_segment\_fraction" olmak üzere ikiye ayrılmaktadır. Bu parametrelerden "range" arama ağacında üretilecek rastgele düğümün maksimum uzaklığını tanımlarken, "longest\_valid\_segment\_fraction" parametresi ise OMPL'nin hareketi doğrulaması için hareket parçalarının uzunluğu ayarlamaya yardımcıdır. Öyle ki, uzunluğu tanımlanmış her bir hareket parçası sonunda OMPL hareketin uygunluğunu kontrol etmektedir. Çalışma alanındaki herhangi bir nesneyle çarpışmanın önlenmesi için bu parametrenin değeri düşük tutulmaktadır.

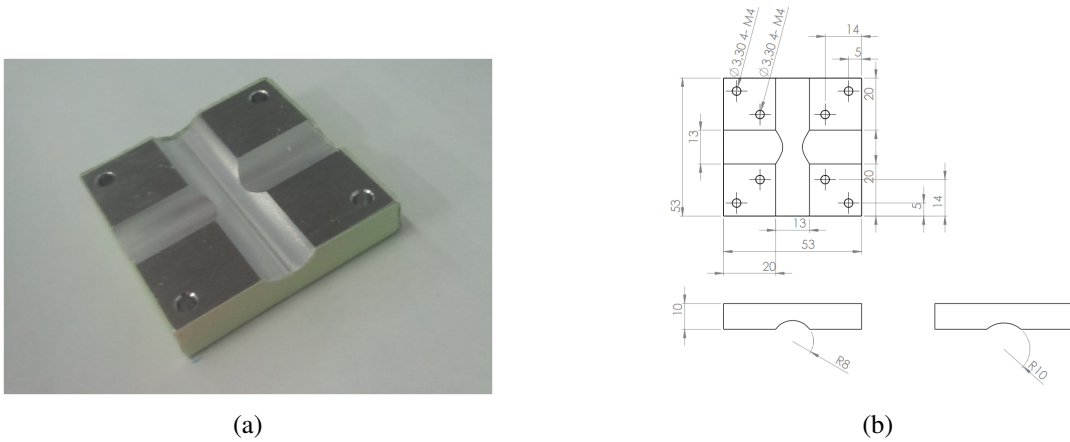
Robotun belirlenen endüstriyel işlemi gerçekleştirebilmesi için yapacağı hareket, sonuç olarak ROS platformu üzerinde ve MoveIt yazılımının içerdiği OMPL kütüphanesi yardımıyla oluşturulan hareket planları ile sağlanmaktadır.

## 5. UYGULAMALAR

Bu tez çalışmasında, öncelikle önerilen sistemin uygulanabilirliğini ve çalışmasını göstermek amacıyla tüm süreçlerin ortaya koyulduğu bir uygulama ve sonrasında oluşturulan hareket planının engellerden sakınacak özellikte olduğunu açık bir şekilde göstermek için bir başka uygulama sunulmaktadır.

### 5.1. Yöntem Testi

Bu uygulama için kullanılan nesne, alüminyumdan üretilmiş ve vida delikleri içeren konkav özellikte endüstriyel bir nesnedir ve Şekil 5.1’te gösterilmektedir. Gerçekleştirilen uygulama, endüstriyel parça üzerinde yer alan ve kullanıcı tarafından tarama sonucunda oluşan görsel üzerinden seçilen vida deliklerinin merkezine, robot kolunun uç kısmına monte edilmiş uç elemanının götürülmesini içermektedir. Bu uygulama, ilk olarak endüstriyel parçanın görselleştirilmesini, daha sonra oluşturulan nokta bulut görseli üzerinden kullanıcının vida delikleri etrafında üçer nokta seçip seçilen noktalardan vida delik merkezlerinin bulunmasını ve son olarak da bu noktalara robotun sahip olduğu uç elemanını yönlendirerek endüstriyel işlemi gerçekleştirmesini içermektedir.

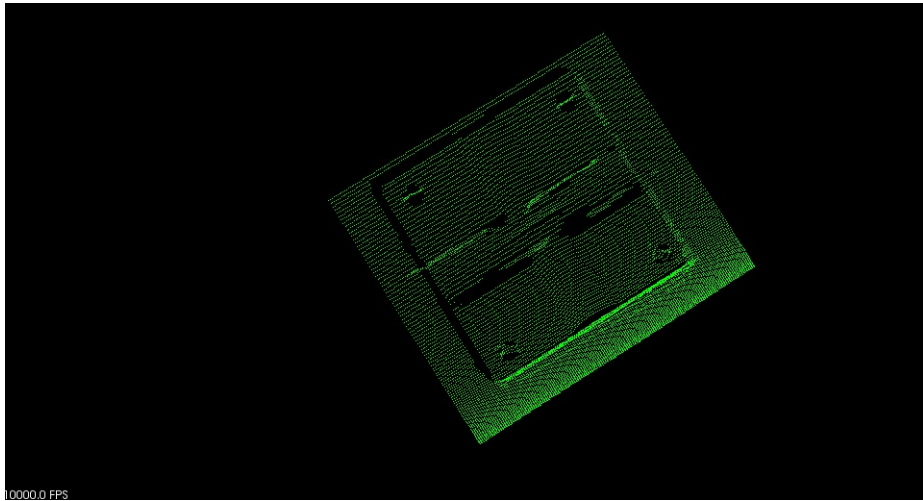


Şekil 5.1: Uygulama için Kullanılan Endüstriyel Parça (a) Görünüş (b) Teknik Çizim

### 5.1.1. Görselleştirme aşaması

Gerçekleştirilen uygulama, ön tanımlı ve robot ucunun lineer bir yol izlediği hareket gerçekleştirilerek endüstriyel parçanın tek bir bakış açısından taranmasıyla başlamaktadır. Taranan endüstriyel parçadan elde edilen 3-B nokta bulutu, daha sonra PCL yardımıyla görselleştirilmektedir.

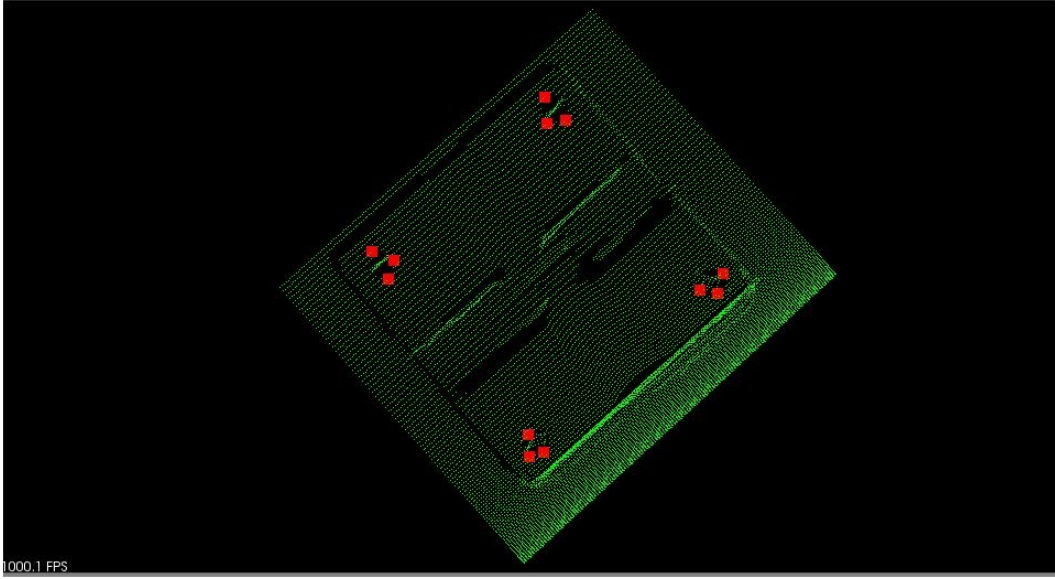
Tarama işlemi için robot denetleyicisi üzerinde yer alan veri toplama amaçlı sunucu yazılımı ile sistem bilgisayarı üzerinde yer alan veri toplama modülü görev almaktadır. Robotun izleyeceği ön tanımlı yörüngede hareket etmesi ve bu hareket esnasında robot ve algılayıcı verilerinin sistem bilgisayarına iletilmesi veri toplama amaçlı sunucu yazılımı sayesinde olmaktadır. Aynı şekilde robot ve algılayıcıdan gönderilen verilerin toplanıp dönüşüm işlemi uygulandıktan sonra bir dosya içerisinde toplanması da veri toplama modülü sayesinde gerçekleştirilmektedir. Tarama işleminden sonra ise dönüşüm işlemi uygulanmış verileri içeren dosya, görselleştirme modülü tarafından kullanılarak görselleştirme işlemi yapılmaktadır. Görselleştirme işlemi sonucunda kullanıcıya bir arayüz üzerinde taranan parçanın 3-boyutlu nokta bulut ifadesi sunulmaktadır. Bu uygulamada taranan parçadan elde edilen 3-boyutlu nokta bulut verisine ait görselleştirme sonucu Şekil 5.2’te gösterilmektedir.



Şekil 5.2: Taranan Parçaya Ait 3-Boyutlu Nokta Bulut Görselleştirilmesi

### 5.1.2. İşlem noktası seçim aşaması

Görselleştirme aşamasının tamamlanmasından sonra kullanıcıya işlem noktası seçmesi için bir arayüz üzerinde taranan parçanın 3-boyutlu nokta bulut modeli sunulmaktadır. Bu model üzerinde kullanıcının seçtiği noktalar kırmızı renkle işaretlenmektedir. Bu işlemlerin her biri, işlem noktası seçme modülü tarafından gerçekleştirilmektedir. Bu uygulamada, görselleştirme işlemi sonucunda oluşturulan 3-boyutlu nokta bulut görseli üzerinde her bir vida deliğinin kenarından üçer nokta seçilmiştir. Görsel üzerinde işlem noktalarının seçilmesi sonucu oluşan görüntü Şekil 5.3’da gösterilmektedir.



Şekil 5.3: Görsel Üzerinden Üçer Adet Nokta Seçilmesine İlişkin Görüntü

İşlem noktalarının 3-boyutlu nokta bulut verisi üzerinden seçilmesinin ardından, bu uygulamaya özel olarak, vida delik merkezlerinin konumları saptanmaktadır. Robotun sahip olduğu uç elemanı hareketi vida delik merkezlerine yönelik gerçekleşeceğinden, seçilen üçer nokta sayesinde büyük çember ve sonrasında büyük çemberin merkezi bulunmaktadır. Bu işlem, bu uygulamada dört vida deliği için de uygulanmış ve dört vida delik merkezinin ko-

numları saptanmıştır. Vida delik merkezlerinin konumlarını saptarken kullanılan algoritma, Şekil 5.4’de gösterilmektedir.

Gerçekleştirilen uygulamada dört adet vida delik merkez koordinatlarının bulunmasının yanında geçiş noktaları da tanımlanmaktadır. Vida deliklerinin bulunduğu yüzeyin belirli bir seviye üzerinde ve vida delik merkezi koordinatlarıyla aynı hizada olacak şekilde tanımlanan geçiş noktalarıyla robotun işlem yapılacak noktaya uç elemanını direkt olarak götürmeden önce geçiş noktalarına götürmesi amaçlanmaktadır. Böylece robot uç elemanının işlem yapılması gerekmeyen noktalara temas edebilecek şekilde yaklaşması olabildiğince önlenmektedir. Gerçekleştirilen uygulamada seçilen noktalar ile birlikte belirlenen vida delik merkezi ve geçiş noktalarının koordinatları Çizelge 5.1’de gösterilmektedir.

Çizelge 5.1: Uygulamada Seçilen Noktalar ile Hesaplanan Delik Merkezi ve Geçiş Noktası Koordinatları

Seçilen Noktaların Koordinatları (X,Y,Z)	Hesaplanan Delik Merkezi Koordinatları (X,Y,Z)	Atanan Geçiş Noktası Koordinatları (X,Y,Z)
155.399 , 608.345 , -279.647		
158.018 , 605.485 , -279.721	157.446 , 607.592 , -279.766	157.446 , 607.592 , -269.766
159.494 , 608.330 , -279.950		
155.177 , 650.975 , -279.200		
156.483 , 648.580 , -279.275	157.309 , 650.591 , -279.459	157.309 , 650.591 , -269.459
159.421 , 651.060 , -279.750		
197.890 , 652.015 , -279.612		
201.944 , 651.130 , -280.255	199.678 , 650.419 , -279.850	199.678 , 650.419 , -269.850
198.848 , 648.170 , -279.628		
202.806 , 607.840 , -280.024		
197.078 , 607.845 , -279.967	199.942 , 608.148 , -279.960	199.942 , 608.148 , -269.960
199.088 , 605.415 , -280.267		

---

**Veri:** Seçilen üç noktanın koordinatları ;

**Sonuç:** Büyük çember merkezinin koordinatları ;

Prosedür;

Seçilen noktaları  $P1, P2, P3$  olarak tanımla;

$P1$  ve  $P2$  arasındaki vektörü hesapla ( $V1$ );

$P1$  ve  $P3$  arasındaki vektörü hesapla ( $V2$ );

$V1$  vektörünün uzunluğu hesapla ( $L1$ );

$V1$  vektörünün uzunluğu hesapla ( $L2$ );

**Eğer**  $L1=0$  veya  $L2=0$  ise;

Özdeş point hatası;

$V1$  vektörünü normalize et ( $VN1$ );

$V2$  vektörünü normalize et ( $VN2$ );

Çemberin düzlemi için normal vektörü hesapla ( $NV$ );

**Eğer**  $NV$  çok küçük bir değer ise;

Kolineer nokta hatası;

$V2N$  vektörünün  $V1N$  üzerindeki ortogonalini bul ( $V2NB$ );

$V2NB$  vektörünün uzunluğu hesapla ( $L2NB$ ) ;

$V2NB$  vektörünü normalize et ( $V2NNB$ ) ;

Çember düzlemini ayırklaştır ;

İlk koordinat vektörünü  $V1N$  olarak belirle ;

İkinci koordinat vektörünü  $V2NB$  olarak belirle ;

Düzlem üzerindeki normal vektörünü  $NV$  olarak belirle ;

Her bir düzlemdeki noktaların 2-B koordinatlarını hesapla ;

2-B çember üzerindeki noktaları saptayıp buna uyan çemberi hesapla ;

Kesişen dikey bisektörleri kullanarak çember merkezini bul ;

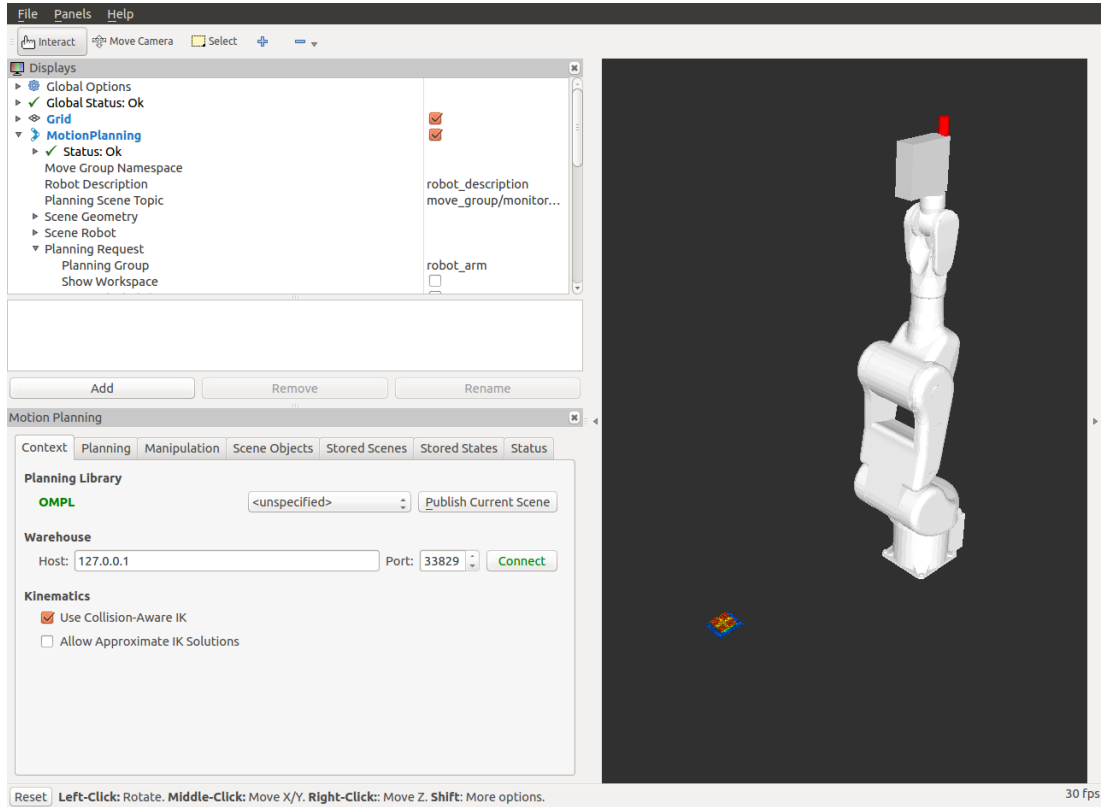
---

Şekil 5.4: Büyük Çember Merkezlerini Saptamak İçin Kullanılan Algoritma

### 5.1.3. İşlem uygulama aşaması

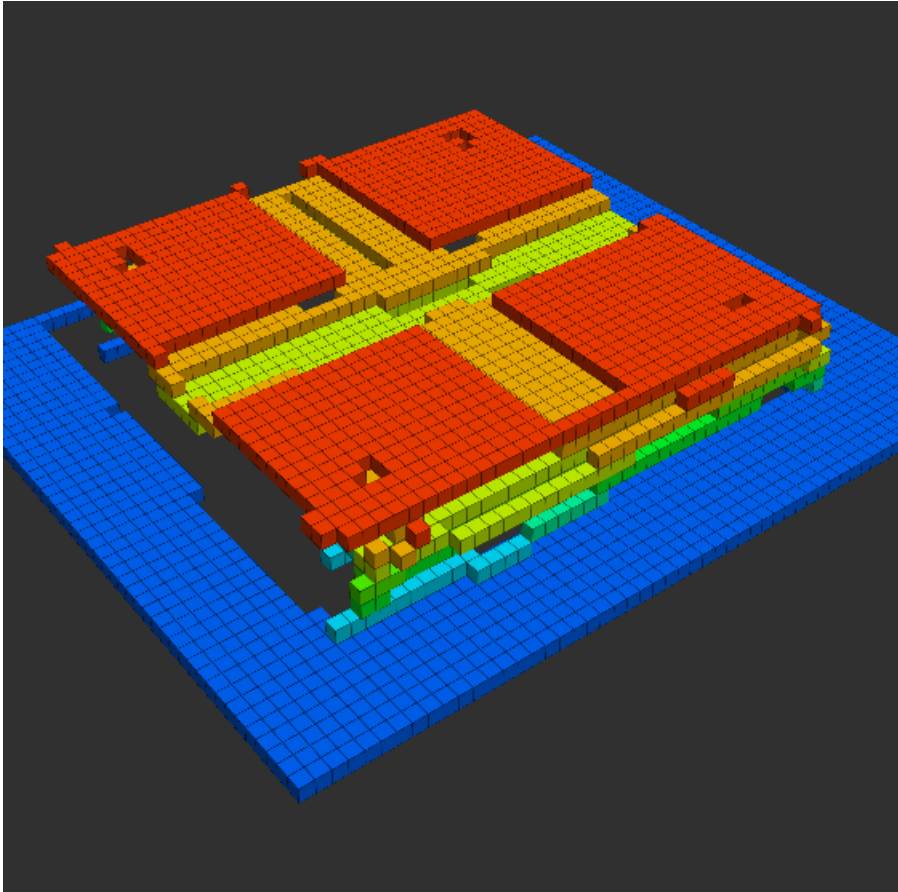
İşlem noktası seçim aşamasının tamamlanmasıyla birlikte işlemin uygulanması aşaması başlamaktadır. Bu uygulamada robot, hesaplanan vida delik merkezi koordinatlarına ve bu noktalarla birlikte tanımlanan geçiş noktalarına sahip olduğu uç elemanını götürerek endüstriyel işlemleri gerçekleştirmektedir.

Bu uygulamadaki hareket planlama sürecinde, MoveIt yazılımı planlama aşaması için OMPL kütüphanesi ile birlikte kullanılmaktadır. Bunun yanında "Rviz" adlı görselleştirme aracına da uygulamanın simüle edilmesi amacıyla başvurulmaktadır. "Rviz" üzerinde, planlama süreci için kullanılan robot kolu modelinin ve 3-B nokta bulut verisinin yer aldığı görüntü Şekil 5.5'de gösterilmektedir.



Şekil 5.5: Simülasyon Ortamında Robot Kolu Modeli ve 3-B Nokta Bulut Verisi

Hareket planlama aşamasının ilk adımında planlama yapılacak ortamın oluşturulması yer almaktadır. Planlama yapılacak ortamın oluşturulması, algılayıcı sayesinde elde edilen 3-B nokta bulut verisi gibi verilerin kullanımıyla sağlanmaktadır ve bu sayede çarpışma kontrolü yapmak mümkün olmaktadır. Herhangi bir algılayıcı ile elde edilebilecek 3-B nokta bulut verisi, MoveIt yazılımı ile "octomap" şeklinde planlama ortamına yüklenebilmektedir. Bu uygulamada ilk olarak algılayıcı verisi planlama ortamına Şekil 5.6'da gösterildiği gibi eklenmiştir.



Şekil 5.6: Planlama Ortamına Yüklenen Nokta Bulut Verisi

Hareket planlamanın bir sonraki aşamasında ise belirlenen vida delik merkezlerinin ve geçiş noktalarının koordinatları robot işlem uygulama modülünün sahip olduğu hareket



planlayıcı tarafından işlenerek bir hareket planı oluşturulmaktadır. İşlem noktası seçim aşaması sonucunda dosyaya kaydedilmiş hareket noktaları Çizelge 5.2’de gösterildiği gibi on üç noktadan oluşan bir nokta dizisi halinde bulunmaktadır ve bu dizide yer alan ilk iki nokta hareket planlaması başlarken sırasıyla başlangıç ve hedef noktaları olarak tanımlanmaktadır. Bu iki nokta arasında hareket planlandıktan sonra hedef olan nokta, sıradaki planlamada başlangıç noktası olmakta ve dizi içerisinde bir sonraki sırada yer alan nokta ise hedef noktası olmaktadır. Bu süreç dizideki son nokta hedef noktası oluncaya kadar sürmektedir. Bu uygulamada da bu süreç yürütülmüş ve devamlı olarak iki nokta arasında hareket planlanarak endüstriyel işlemin gerçekleştirilmesi sağlanmıştır. Hareket planlama süreci, OMPL kütüphanesinde yer alan RRT-Connect algoritmasının MoveIt yazılımı ile birlikte kullanılmasıyla gerçekleştirilmiştir. Simülasyon ortamında endüstriyel işlemin gerçekleştirilmesi esnasında sırasıyla alınan görüntüler Şekil 5.7’de gösterilmektedir.

Çizelge 5.2: Hareket Planlamasının Gerçekleştirildiği Nokta Dizisi

---

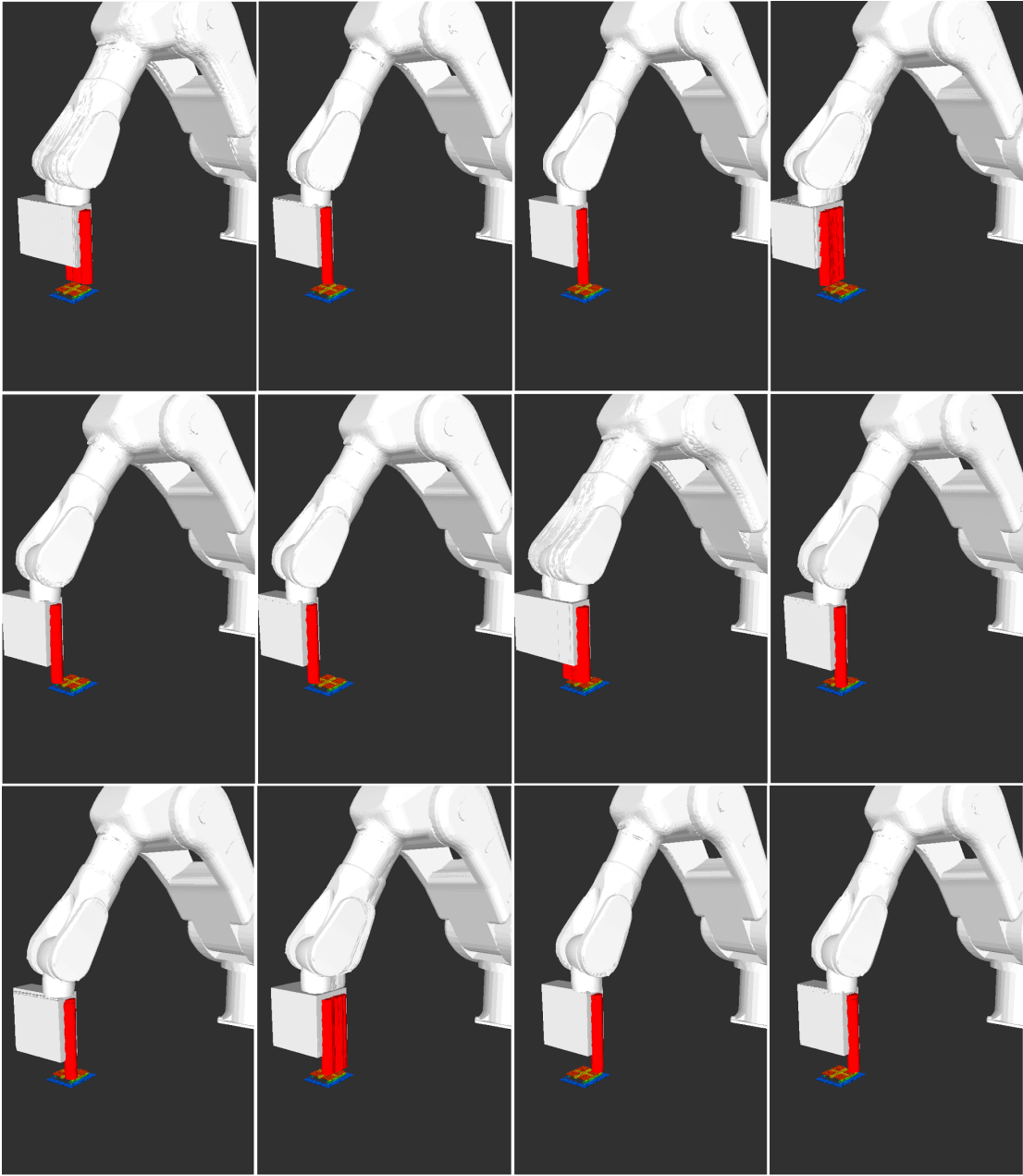


---

Hareket Planlama için Kullanılacak Noktaların Koordinatları (X,Y,Z)

1. Nokta	199.942 , 608.148 , -269.960
2. Nokta	157.446 , 607.592 , -269.766
3. Nokta	157.446 , 607.592 , -279.766
4. Nokta	157.446 , 607.592 , -269.766
5. Nokta	157.309 , 650.591 , -269.459
6. Nokta	157.309 , 650.591 , -279.459
7. Nokta	157.309 , 650.591 , -269.459
8. Nokta	199.678 , 650.419 , -269.850
9. Nokta	199.678 , 650.419 , -279.850
10. Nokta	199.678 , 650.419 , -269.850
11. Nokta	199.942 , 608.148 , -269.960
12. Nokta	199.942 , 608.148 , -279.960
13. Nokta	199.942 , 608.148 , -269.960

---



Şekil 5.7: Simülasyon Ortamında Endüstriyel İşlemin Gerçekleştirildiği Süreçte Sırayla Alınan Görüntüler

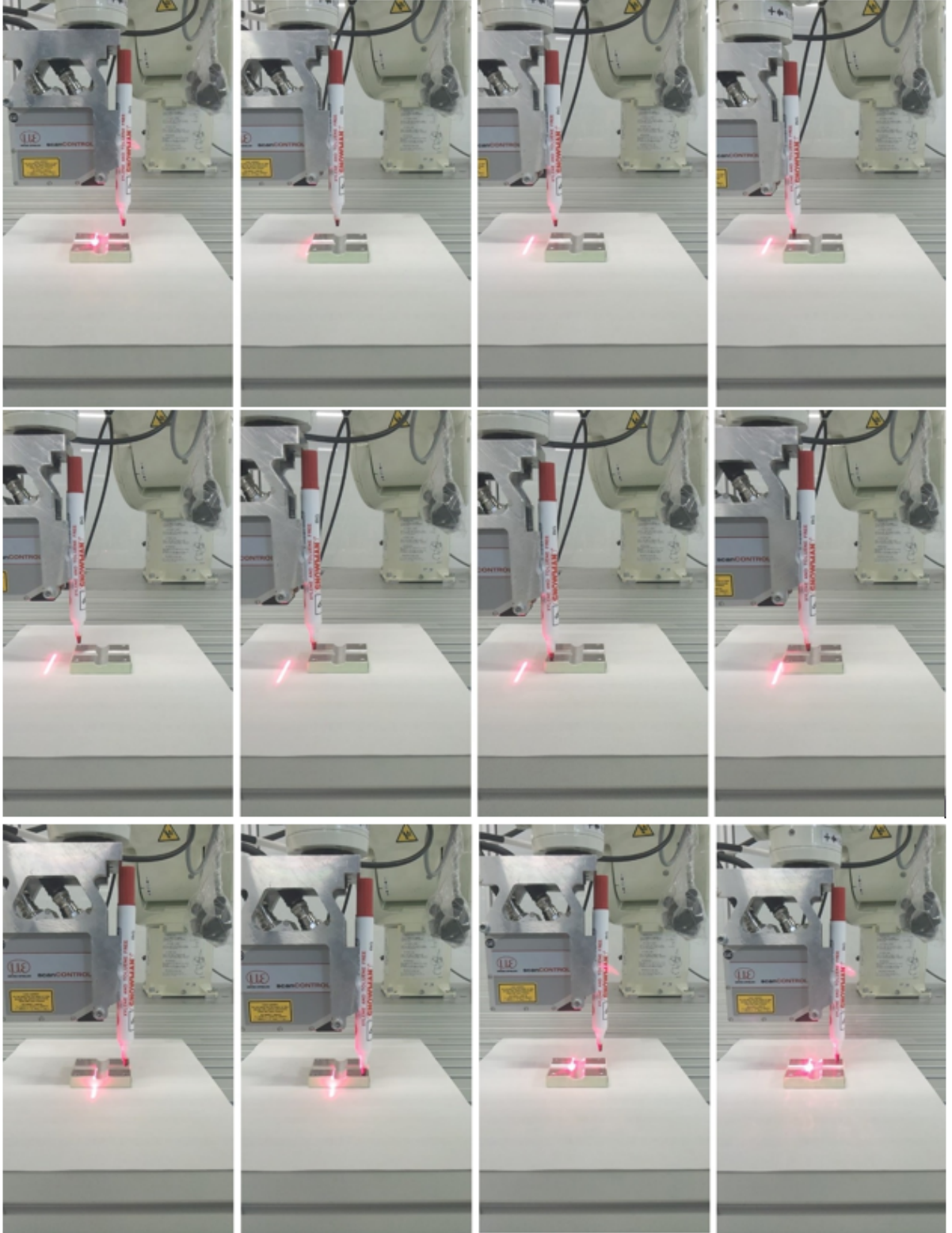
Bu tez çalışmasında gerçekleştirilen uygulamada planlayıcı olarak RRT-Connect algoritması kullanılmıştır. RRT-Connect algoritması yapısı gereği başlangıç ve hedef konumlarından durumlar üreterek çözüme ulaşmaya çalışmaktadır. Bu uygulamada da, iki nokta arasındaki her bir planlama parçası için bu durum gerçekleşmiştir. Her bir planlama parçasında çözüme ulaşırken üretilen durum sayısı ve çözümün bulunma süresi Çizelge 5.3'de

gösterilmektedir. RRT-Connect algoritması kullanımıyla uygulama sonucunda elde edilen yörünge için toplamda 53 durum oluşturulmuş ve endüstriyel işlemi gerçekleştirecek çözümlerin bulunması ise toplamda 0.280688 saniye sürmüştür.

Çizelge 5.3: RRT-Connect Algoritmasının Planlama Parçalarında Üretilen Durum Sayısı ve Çözüme Ulaşma Süresi

Planlama Parçası	Üretilen Durum Sayısı	Çözüme Ulaşma Süresi
1.Nokta-2.Nokta için Planlama	4 durum (2 başlangıç + 2 hedef)	0.022391 saniye
2.Nokta-3.Nokta için Planlama	5 durum (3 başlangıç + 2 hedef)	0.026648 saniye
3.Nokta-4.Nokta için Planlama	4 durum (2 başlangıç + 2 hedef)	0.026067 saniye
4.Nokta-5.Nokta için Planlama	4 durum (2 başlangıç + 2 hedef)	0.042198 saniye
5.Nokta-6.Nokta için Planlama	5 durum (2 başlangıç + 3 hedef)	0.026603 saniye
6.Nokta-7.Nokta için Planlama	4 durum (2 başlangıç + 2 hedef)	0.023730 saniye
7.Nokta-8.Nokta için Planlama	4 durum (2 başlangıç + 2 hedef)	0.024492 saniye
8.Nokta-9.Nokta için Planlama	4 durum (2 başlangıç + 2 hedef)	0.022028 saniye
9.Nokta-10.Nokta için Planlama	5 durum (2 başlangıç + 3 hedef)	0.027695 saniye
10.Nokta-11.Nokta için Planlama	5 durum (2 başlangıç + 2 hedef)	0.035484 saniye
11.Nokta-12.Nokta için Planlama	4 durum (2 başlangıç + 2 hedef)	0.025870 saniye
12.Nokta-13.Nokta için Planlama	5 durum (2 başlangıç + 3 hedef)	0.025940 saniye

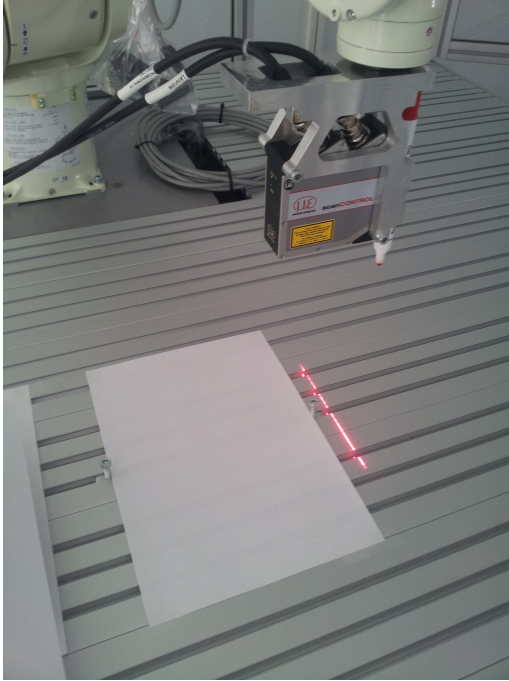
Hareket planlama aşamasının sonunda oluşan hareket planı ise, robotun oluşan yörüngeyi takip ederken belirli zaman aralıklarında sahip olduğu eklem açılarının robot denetleyicisinde var olan işlem uygulama amaçlı sunucu yazılımı tarafından alınmasıyla gerçekleştirilmektedir. Gerçekleştirilen uygulamada dört adet vida delik merkezini ve geçiş noktalarının koordinatlarıyla oluşturulan hareket planı işlem uygulama amaçlı sunucu yazılımı tarafından kullanılarak robotun endüstriyel işlemi gerçekleştirmesi sağlanmıştır. Robotun endüstriyel işlemi gerçekleştirmesi esnasında sırasıyla alınan görüntüler Şekil 5.8'de gösterilmektedir.



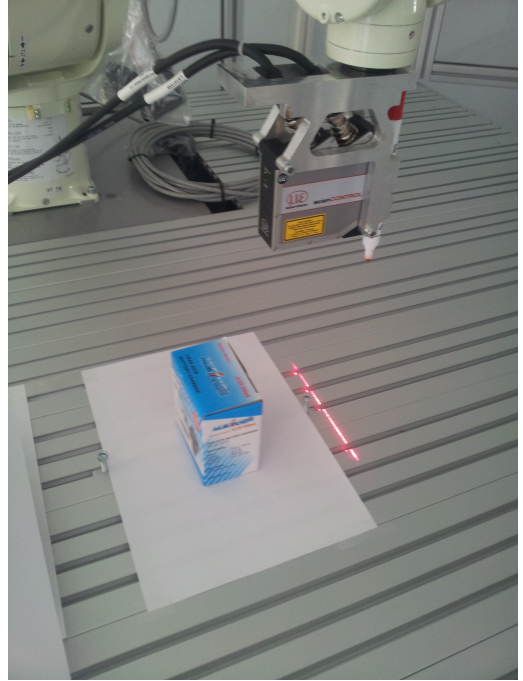
Şekil 5.8: Endüstriyel İşlemin Gerçekleştirildiği Süreçte Sırayla Alınan Görüntüler

## 5.2. Hareket Planlama Testi

Bu uygulama için oluşturulan ortam, Şekil 5.9'de gösterildiği üzere çalışma uzayı içerisinde iş istasyonu zeminine sabitlenmiş iki vida içermektedir. Bunun yanında oluşturulan hareket planının engellerden sakınan özellikte olduğunu göstermek amacıyla oluşturulan bu uygulamanın bir senaryosunda iki vida arasına herhangi bir engel konulmazken, bir diğer senaryosunda iki vida arasına engel olarak işlev görecektir bir kutu konulmaktadır.



(a)



(b)

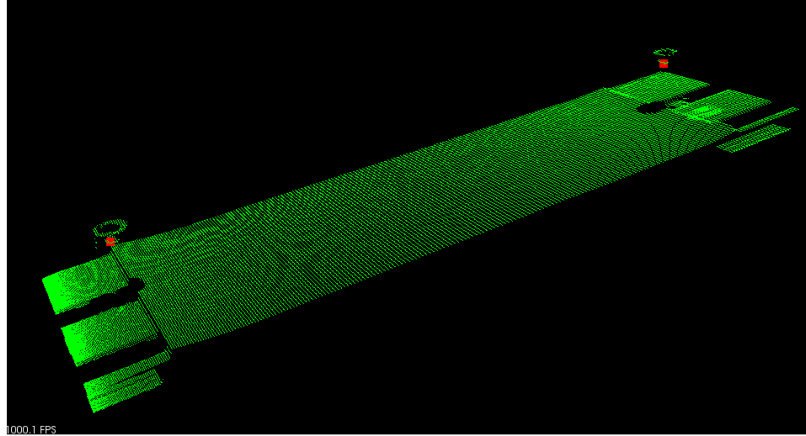
Şekil 5.9: Hareket Planlama Testi İçin Ortaya Konulan Uygulama Ortamları (a) Engel İçermeyen (b) Engel İçeren

### 5.2.1. Görselleştirme aşaması ve işlem noktası seçim aşaması

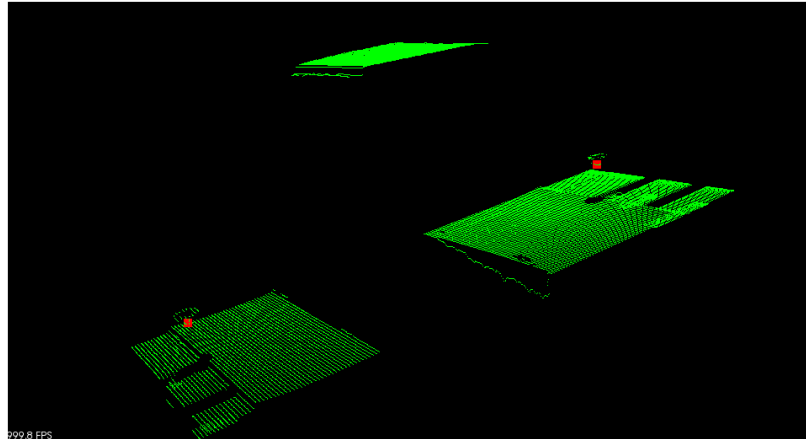
Gerçekleştirilen uygulama, ön tanımlı ve robot ucunun lineer bir yol izlediği hareket gerçekleştirilerek oluşturulan ortamların tek bir bakış açısından taranmasıyla başlamaktadır.

Taranan ortamlardan elde edilen 3-B nokta bulutu, daha sonra PCL yardımıyla görselleştirilmiştir. Görselleştirme işlemi sonucunda kullanıcıya bir arayüz üzerinde taranan ortamların 3-boyutlu nokta bulut ifadeleri sunulmaktadır.

Görselleştirme aşamasının tamamlanmasından sonra kullanıcıya işlem noktası seçmesi için bir arayüz üzerinde taranan ortamların 3-boyutlu nokta bulut modelleri sunulmaktadır. Bu modeller üzerinde kullanıcının seçtiği noktalar kırmızı renkle işaretlenmektedir. Bu uygulamada taranan ortamlardan elde edilen 3-boyutlu nokta bulut verilerine ait görselleştirme sonuçları ve görseller üzerinde işlem noktalarının seçilmesi sonucu oluşan görüntüler Şekil 5.10'te gösterilmektedir.



(a)



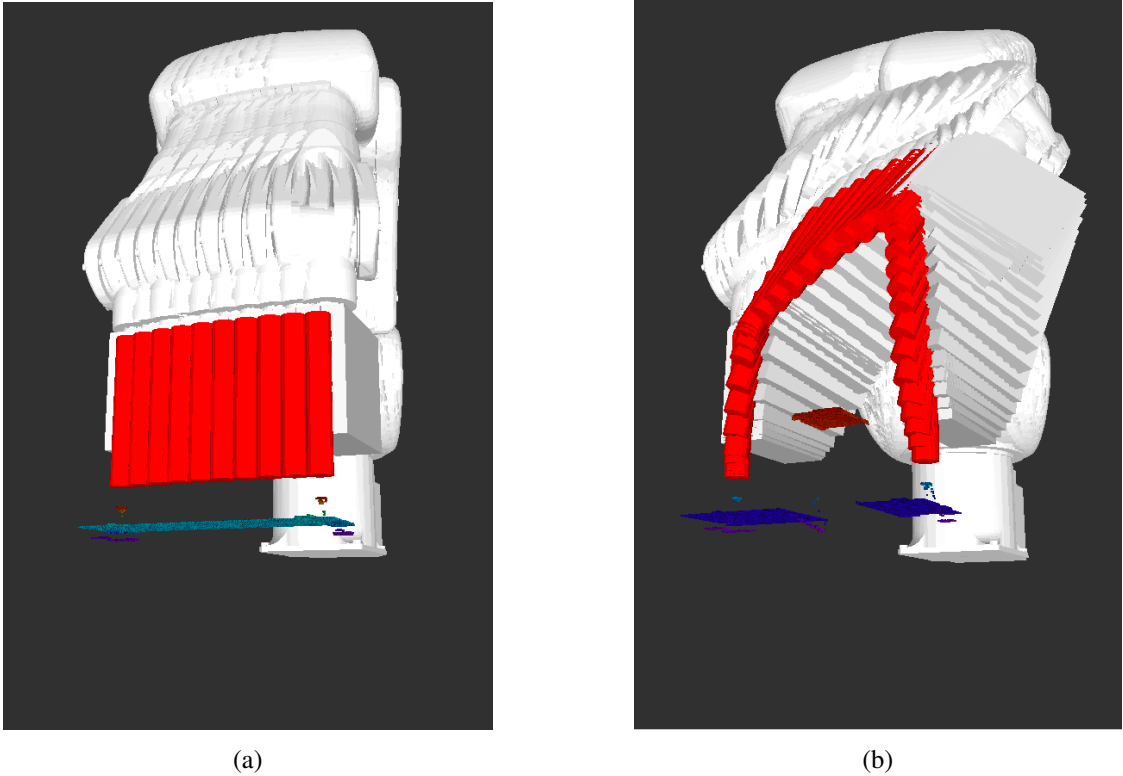
(b)

Şekil 5.10: Hareket Planlama Testi İçin Elde Edilen Görselleştirme Sonuçları (a) Engel İçermeyen (b) Engel İçeren

### 5.2.2. İşlem uygulama aşaması

İşlem noktası seçim aşamasının tamamlanmasıyla birlikte işlemin uygulanması aşaması başlamaktadır. Bu uygulamada robot, oluşturulan iki senaryoda da kullanıcı tarafından seçilen vida delik merkezi koordinatlarının belirli bir seviye üzerine sahip olduğu uç elemanını götürerek endüstriyel işlemleri gerçekleştirmektedir.

Bu uygulamada da hareket planlama süreci, OMPL kütüphanesinde yer alan RRT-Connect algoritmasının MoveIt yazılımı ile birlikte kullanılmasıyla gerçekleştirilmiştir. Simülasyon ortamında hareket planlama işleminin gerçekleştirilmesi esnasında engel içeren ve engel içermeyen ortam için alınan görüntüler Şekil 5.11’de gösterilmektedir.



Şekil 5.11: Hareket Planlama Testi İçin Oluşturulan Hareket Planları (a) Engel İçermeyen (b) Engel İçeren

Hareket planlama aşamasının sonunda oluşan hareket planı ise, yöntem testinde olduğu gibi robotun oluşan yörüngeyi takip ederken belirli zaman aralıklarında sahip olduğu eklem açılarının robot denetleyicisinde var olan işlem uygulama amaçlı sunucu yazılımı tarafından alınmasıyla gerçekleştirilmektedir. Robotun, engel içeren hareket planlama testi senaryosunu gerçekleştirmesi esnasında sırasıyla alınan görüntüler Şekil 5.12’de gösterilmektedir.



Şekil 5.12: Hareket Planlama Testinin Engel İçeren Senaryosunun Gerçekleştirildiği Süreçte Sırayla Alınan Görüntüler



## 6. SONUÇLAR

### 6.1. Sonuçlar

Bu çalışmada, endüstriyel robot kolu ve lazer profil algılayıcı yardımıyla taranan ve 3-boyutlu nokta bulut verisi elde edilen endüstriyel nesne üzerinde otonom olarak endüstriyel bir işlem gerçekleştirilmesi amaçlanmıştır. Bu amaca yönelik olarak taranan bir nesneye ait görsel üzerinden kullanıcıya işlem yapılacak noktaları seçme imkanı tanınmıştır. İşlem yapılmak üzere kullanıcı tarafından belirlenen noktalar kullanılarak ise hareket planı oluşturulup robot kolunun bu hareketi takip etmesi sağlanmıştır.

Oluşturulan donanım, 6-eksenli endüstriyel robot kolu ve lazer profil algılayıcı içermektedir. Sahip olunan robot kolunun yüksek doğruluğu ve lazer profil algılayıcının yüksek çözünürlüğü, robotun örnek uygulamada gerçekleştirilen endüstriyel işlemi başarılı bir şekilde tamamlamasını sağlamıştır. Aynı zamanda algılayıcının sunduğu farklı konfigürasyon ayarları da elde edilen 3-boyutlu nokta bulutu konusunda esneklik sağlamıştır.

Robot denetleyicisi ile lazer profil algılayıcının, sistem bilgisayarını ile iletişimi TCP/IP bağlantısı aracılığıyla sağlanmıştır. Soket programlama içeren yazılımlar sayesinde yazılım modüllerinin birbirleriyle TCP/IP bağlantısı üzerinden iletişimi sorunsuz bir şekilde gerçekleştirilmiştir.

Ortaya koyulan örnek uygulama da sahip olunan donanım ve geliştirilen yazılımın, örnek uygulamadaki gibi bir endüstriyel işlemi başarılı bir şekilde gerçekleştirebileceğini göstermiştir. Endüstriyel robot kolları için otomatik programlama yöntemi geliştirilmesine yönelik yapılan bu tez çalışmasıyla bu türden sistemlerin fizibilitesi ortaya koyulmuş ve alt yapısı sunulmuştur.

Bu tez çalışması süresince gerçekleştirilen çalışmalar aynı zamanda ulusal ve uluslararası konferanslarda sunulmuştur (Seçil vd., 2014 a; Seçil vd., 2014 b).

## 6.2. İleriye Yönelik Çalışmalar

Bu tez çalışmasında önerilen sistemle endüstriyel robot kolları için otomatik programlama yöntemi geliştirilmesine yönelik olarak bir altyapı ortaya koyulmuştur. Sahip olunan yazılım bu aşamada otomatik programlama yönteminin temelini içerse de, otonom bir süreç gerçekleşmemektedir. İleriye yönelik yapılacak çalışmalarla bu sürecin tam otonom hale getirilmesi planlanmaktadır.

İleriye yönelik çalışmalarda dikkate alınacak bir diğer husus da yapılan ölçümlerin doğruluğunu saptamaktır. Gerekli ölçüm sistemlerinin temin edilmesiyle beraber yapılan ölçümlerdeki hata payı ortaya koyulacak ve bu hataların endüstriyel işlemin gerçekleştirilmesine nasıl bir etki ettiği tartışılacaktır.

Elde edilen nokta bulut verisinden katı cisim modelleri oluşturabilecek şekilde nesnelerin taranması, taranmayan noktaların saptanması ve bu bölgelerin taranması için geliştirilecek yöntemler, 3-B veri toplama amacıyla kullanılacak uygun hareket planlama yöntemlerinin ortaya konulması, sahip olunan sistemin başka türden elemanlarla (döner tabla, derinlik sensörü vb.) entegre edilmesi ileriye yönelik yapılması planlanan diğer çalışmalardır.

## KAYNAKLAR DİZİNİ

- Brooks, R. A. ve Perez, T. L., 1983. A subdivision algorithm in configuration space for findpath with rotation, In Proceedings of the 8th International Conference on Artificial Intelligence, pages 799–906.
- Bruccoleri, M., D’Onofrio, C. ve Commare, U. L., 2007. Off-line programming and simulation for automatic robot control software generation, 5th IEEE International Conference on Industrial Informatics, 1(3):491–496.
- Chan, V., Bradley, C. ve Vickers, G., 2000. A multi-sensor approach for rapid digitization and data segmentation in reverse engineering, Journal of Manufacturing Science and Engineering, 122:725–733.
- Chazelle, B., 1987. Approximation and decomposition of shapes, Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics, pages 145–185.
- Jeon, J. W., Park, S.-H., Kim, D.-I. ve Kim, S., 1993. An efficient trajectory generation for industrial robots, Industry Applications Society Annual Meeting, 3:2137–2143.
- Karbacher, S., Laboureux, X., Schon, N. ve Hausler, G., 2001. Processing range data for reverse engineering and virtual reality, Third International Conference On 3-D Digital Imaging and Modeling, pages 314–321.
- Kavraki, L. E., Svestka, P., Latombe, J. ve Overmars, M. H., 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Transactions on Robotics and Automation, 12(4):566–580.
- Kawasaki Heavy Industries, 2015. Kawasaki robot controller e series as language reference manual, <http://robotics.kawasaki.com>, erişim tarihi: 30.05.2015.
- Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots, The International Journal of Robotics Research, 5(1):90–98.

**KAYNAKLAR DİZİNİ (devam ediyor)**

- Lanman, D., Crispell, D. ve Taubin, G., 2009. Surround structured lighting: 3-d scanning with orthographic illumination, *Computer Vision and Image Understanding*, 113:1107–1117.
- Larsson, S. ve JAP, K., 2006. Motion control and data capturing for laser scanning with an industrial robot, *Robotics and Autonomous Systems*, 54(6):453–460.
- Larsson, S. ve Kjellander, J., 2007. Path planning for laser scanning with an industrial robot, *Robotics and Autonomous Systems*, 56:615–624.
- LaValle, S. M., 1998. Rapidly-exploring random trees: A new tool for path planning, Technical report, Technical Report 98-11, Computer Science Dept., Iowa State University.
- Luyi, C., 2011. A novel method for workpiece orientation in robot glueing system, *Proceedings of 2011 International Conference on Instrumentation, Measurement, Circuits and Systems*, 3:357–360.
- Micro-Epsilon, 2015. Scancontrol 2900-100 laser profile scanner contents and documentation, <http://www.micro-epsilon.com>, erişim tarihi: 30.05.2015.
- Nayak, N., Ray, A. ve Vavreck, A. N., 2003. An adaptive real-time intelligent seam tracking system, *Journal of Manufacturing Systems*, 6(3):241–245.
- Papadopoulos, G., Kurniawati, H., Shariff, A., Wong, L. ve N.M.Patrikalakis, 2011. 3d-surface reconstruction for partially submerged marine structures using an autonomous surface vehicle, *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3551–3557.
- Seçil, S., Turgut, K., Özkan, M. ve Parlaktuna, O., 2014a. 3-d visualization system for geometric parts using a laser profile sensor and an industrial robot.
- Seçil, S., Turgut, K., Özkan, M. ve Parlaktuna, O., 2014b. Laser profil algılayıcı ve endüstriyel robot kullanımı ile geometrik parçalar için 3-b görselleştirme sistemi.

**KAYNAKLAR DİZİNİ (devam ediyor)**

- Son, S., Kim, S. ve Lee, K. H., 2003. Path planning of multi-patched freeform surfaces for laser scanning, *Int J Adv Manuf Technol*, 22:424–435.
- Son, S., Park, H. ve Lee, K. H., 2002. Automated laser scanning system for reverse engineering and inspection, *International Journal of Machine Tools & Manufacture*, 42:889–897.
- Sucan, I. A. ve Chitta, S., 2013. Moveit system architecture, <http://moveit.ros.org/documentation/concepts/>, erişim tarihi: 30.05.2015.
- Tognola, G., Parazzini, M., Ravazzani, P., Grandori, F. ve Svelto, C., 2003. 3-d acquisition and quantitative measurements of anatomical parts by optical scanning and image reconstruction from unorganized range data, *IEEE Transactions on Instrumentation and Measurement*, 52:1665–1673.
- van den Berg, J. P. ve Overmars, M. H., 2005. Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners, *International Journal of Robotics Research*, 24(12):1055–1071.
- Vincenzo, N., Cesare, R. ve Sergio, S., 2010. A new real-time shape acquisition with a laser scanner: First test results, *Robotics Computer-Integrated Manufacturing*, 26:543–550.
- Wurll, C. ve Henrich, D., 2001. Point-to-point and multi-goal path planning for industrial robots, Special Issue on "Motion Planning" of the *Journal of Robotic Systems*.
- Xie, Z., Zhang, C., Zhang, Q. ve Zhang, G., 2005. Modelling and verification of a five-axis laser scanning system, *International Journal of Advanced Manufacturing Technology*, 26:391–398.
- Zexiao, X., Jianguo, W. ve Ming, J., 2007. Study on a full field of view laser scanning system, *International Journal of Machine Tools Manufacture*, 47:33–43.

**EK AÇIKLAMALAR-A****ROBOT URDF MODEL DOSYASI**

```
<?xml version="1.0"?>
<robot name="RS05L">
  <link name="base_link">
    <visual>
      <geometry>
        <mesh filename="package://meshes/RS005L_J0_a.dae"/>
      </geometry>
      <material name="gray">
<color rgba="0.7 0.7 0.7 1.0"/>
      </material>
    </visual>
    <collision>
      <geometry>
        <mesh filename="package://meshes/RS005L_J0_a.dae"/>
      </geometry>
    </collision>
  </link>

  <link name="link1">
    <visual>
      <geometry>
        <mesh filename="package://meshes/RS005L_J1_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 285"/>
      <material name="gray">
<color rgba="0.7 0.7 0.7 1.0"/>
      </material>
    </visual>
    <collision>
      <geometry>
        <mesh filename="package://meshes/RS005L_J1_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 285"/>
    </collision>
  </link>

  <joint name="base_to_link1" type="revolute">
```

```

    <axis xyz="0 0 1"/>
    <limit effort="30" velocity="1.0" lower="-3.14" upper="3.14"/>
    <parent link="base_link"/>
    <child link="link1"/>
</joint>

<link name="link2">
  <visual>
    <geometry>
      <mesh filename="package://meshes/RS005L_J2_a.dae"/>
    </geometry>
    <origin rpy="0 0 0" xyz="0 0 0"/>
    <material name="gray">
<color rgba="0.7 0.7 0.7 1.0"/>
    </material>
  </visual>
  <collision>
    <geometry>
      <mesh filename="package://meshes/RS005L_J2_a.dae"/>
    </geometry>
    <origin rpy="0 0 0" xyz="0 0 0"/>
  </collision>
</link>

<joint name="link1_to_link2" type="revolute">
  <axis xyz="1 0 0"/>
  <origin rpy="0 0 0" xyz="0 105 285"/>
  <limit effort="30" velocity="1.0" lower="-2.36" upper="1.4"/>
  <parent link="link1"/>
  <child link="link2"/>
</joint>

<link name="link3">
  <visual>
    <geometry>
      <mesh filename="package://meshes/RS005L_J3_a.dae"/>
    </geometry>
    <origin rpy="0 0 0" xyz="0 0 0"/>
    <material name="gray">
<color rgba="0.7 0.7 0.7 1.0"/>
    </material>
  </visual>
  <collision>
    <geometry>
      <mesh filename="package://meshes/RS005L_J3_a.dae"/>
    </geometry>
    <origin rpy="0 0 0" xyz="0 0 0"/>

```

```

    </collision>
  </link>

  <joint name="link2_to_link3" type="revolute">
    <axis xyz="1 0 0"/>
    <origin rpy="0 0 0" xyz="0 0 380"/>
    <limit effort="30" velocity="1.0" lower="-3" upper="2.06"/>
    <parent link="link2"/>
    <child link="link3"/>
  </joint>

  <link name="link4">
    <visual>
      <geometry>
        <mesh filename="package://meshes/RS005L_J4_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 0"/>
      <material name="gray">
<color rgba="0.7 0.7 0.7 1.0"/>
      </material>
    </visual>
    <collision>
      <geometry>
        <mesh filename="package://meshes/RS005L_J4_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 0"/>
    </collision>
  </link>

  <joint name="link3_to_link4" type="revolute">
    <axis xyz="0 0 1"/>
    <origin rpy="0 0 0" xyz="0 -80 143"/>
    <limit effort="30" velocity="1.0" lower="-6.28" upper="6.28"/>
    <parent link="link3"/>
    <child link="link4"/>
  </joint>

  <link name="link5">
    <visual>
      <geometry>
        <mesh filename="package://meshes/RS005L_J5_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 0"/>
      <material name="gray">
<color rgba="0.7 0.7 0.7 1.0"/>
      </material>
    </visual>

```



```

    <collision>
      <geometry>
        <mesh filename="package://meshes/RS005L_J5_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 0"/>
    </collision>
  </link>

  <joint name="link4_to_link5" type="revolute">
    <axis xyz="1 0 0"/>
    <origin rpy="0 0 0" xyz="0 0 267"/>
    <limit effort="30" velocity="1.0" lower="-2.53" upper="2.53"/>
    <parent link="link4"/>
    <child link="link5"/>
  </joint>

  <link name="link6">
    <visual>
      <geometry>
        <mesh filename="package://meshes/RS005L_J6_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 0"/>
      <material name="black">
<color rgba="0 0 0 1"/>
      </material>
    </visual>
    <collision>
      <geometry>
        <mesh filename="package://meshes/RS005L_J6_a.dae"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 0"/>
    </collision>
  </link>

  <joint name="link5_to_link6" type="revolute">
    <axis xyz="0 0 1"/>
    <origin rpy="0 0 0" xyz="0 0 78"/>
    <limit effort="30" velocity="1.0" lower="-6.28" upper="6.28"/>
    <parent link="link5"/>
    <child link="link6"/>
  </joint>

  <link name="sensor">
    <visual>
      <geometry>
        <box size="52 130 130"/>
      </geometry>

```

```

    <origin rpy="0 0 0" xyz="0 0 0"/>
    <material name="gray">
<color rgba="0.0 0.8 0.0 1.0"/>
    </material>
</visual>
<collision>
    <geometry>
        <box size="52 130 130"/>
    </geometry>
    <origin rpy="0 0 0" xyz="0 0 0"/>
</collision>
</link>

<joint name="link6_to_sensor" type="fixed">
    <axis xyz="0 0 1"/>
    <origin rpy="0 0 0" xyz="0 42.5 65"/>
    <parent link="link6"/>
    <child link="sensor"/>
</joint>

<link name="probe">
    <visual>
        <geometry>
            <cylinder length="160" radius="13"/>
        </geometry>
        <origin rpy="0 0 0" xyz="0 0 0"/>
        <material name="red">
<color rgba="0.8 0.0 0.0 1.0"/>
        </material>
    </visual>
    <collision>
        <geometry>
            <cylinder length="160" radius="13"/>
        </geometry>
        <origin rpy="0 0 0" xyz="0 0 0"/>
    </collision>
</link>
<joint name="sensor_to_probe" type="fixed">
    <axis xyz="0 0 1"/>
    <origin rpy="0 0 0" xyz="0 -69 25"/>
    <parent link="sensor"/>
    <child link="probe"/>
</joint>
</robot>

```